NR INDEKSU 353965
PL ISSN 0860-1674

8

Z MIKROKOMPUTEREM NA TY

Bajtek

MAGAZYN KOMPUTEROWY

NR 8 (66) '91 CENA 10 000 zł

TEMAT MIESIĄCA:
BAZY DANYCH



GRY • NAUKA • MUZYKA • PROGRAMOWANIE

TESTY:

Modem M-2412 PC
Stacja 5.25" do ST
Faxmodem do AMIGI
AMIGA 500
SAM Coupé

Konkurs „7 PYTAŃ" znów nagrody za ponad 4 miliony



PRZEPRASZAMY!

Gwałtowność z jaką podwyżka kursu dolara i podwyżka cen paliw i energii odbiła się na kosztach papieru i usług poligraficznych zaskoczyła nas na tyle, że nie byliśmy w stanie wcześniej uprzedzić Państwa o koniecznym wzroście cen naszego pisma. Staraliśmy się, aby Bajtek kosztował jak najtaniej. W zeszłym roku, zwiększając objętość Bajtka o 25%, udało się nam zachować poprzednią cenę. Trudniej było w tym roku. Przepraszamy, ale mamy jednak nadzieję, że nasze starania o dostarczenie ciekawego i coraz lepszego pisma, przynajmniej częściowo zrekompensują jego wyższą cenę.

Na początku tego roku pisałem o bogatej ofercie sprzętu komputerowego klasy IBM PC możliwego do zakupu przez instytucje i firmy, zwracając jednocześnie uwagę na ubóstwo podobnej oferty, jeśli chodzi o rynek indywidualny. Powoli, ale systematycznie sytuacja w tej dziedzinie zmienia się na korzyść. Powstaje coraz więcej mniejszych lub większych firm zainteresowanych dystrybucją popularnego sprzętu komputerowego dla prywatnych odbiorców. Z satysfakcją odnotowujemy też rosnące zainteresowanie większości, zarówno sprzedawców jak i kupujących, możliwością nabycia legalnego oprogramowania.

Efekty te stają się również widoczne w „Bajtku". Mamy coraz lepszy dostęp do ciekawego sprzętu i programów, których opisy zamieszczamy na naszych łamach, starając się dostarczyć Państwu informacji ułatwiającej dokonanie optymalnych zakupów.

W tym numerze prezentujemy test SAMa Coupe, będącego komputerem 8-bitowym, o możliwościach zbliżonych do Amigi, wyposażonym w stację dysków, ale o cenie porównywalnej z magnetofonowymi zestawami małego Atari lub C64. Osoby posiadające większe zasoby gotówkowe mogą zainteresować się opisem Amigi 500, cieszącej się ostatnio coraz większą popularnością. O tym, że jest to sprzęt mogący znaleźć bardziej profesjonalne zastosowanie, świadczy test faxmodemu do tego komputera.

Klan Atari prezentuje opis rozszerzeń, o jakie można wzbogacić poczciwe ST 520: 4MB RAM, stacja 5.25" i prawie stereofoniczne możliwości dźwiękowe. Nie zapomnieliśmy też o posiadaczach komputerów IBM PC. Modem w formie karty do peceta pozwoli na założenie własnego BBS-u lub wymianę plików i informacji z innymi użytkownikami.

Tematem wiodącym tego numeru są bazy danych. Na kilku stronach staraliśmy się przybliżyć w popularnej formie zagadnienie związane z wykorzystaniem komputera do przechowywania i przetwarzania dużych ilości danych w uporządkowanej formie. Oprócz ogólnego artykułu, prawie w każdym Klanie, znajdą Państwo informacje o programach typu baza danych. Opisywany przez Marcina Borkowskiego pakiet Turbo Data Base został zastosowany do napisania programu mającego usprawnić prenumeratę naszych czasopism. Zobaczymy w niedalekiej przyszłości jakie będą tego efekty.

W Klanie Spectrum, poza wspomnianym wcześniej testem SAMa, znajduje się też pierwszy odcinek artykułu mającego przybliżyć posiadaczom stacji FDD 3000 wykorzystanie jej w systemie CP/M. W Katedrze Nonsensu mogą Państwo znaleźć informacje o nonsensach polityki stosowanej przez naszą pocztę wobec posiadaczy modemów.

Zapraszam do lektury numeru, mając nadzieję, że jego dokładne przeczytanie zwiększy szansę poprawnych odpowiedzi w naszym stałym konkursie „7 pytań" i umożliwi Państwu wylosowanie cennych nagród.

Jarosław Młodzki

Bajtek
MAGAZYN KOMPUTEROWY

Redakcja:
ul. Wspólna 61,
00-687 Warszawa,
tel. 21-12-05

Zespół redakcyjny:
Redaktor Naczelny — Jarosław Młodzki
Sekretarz redakcji — Michał Szokoło
Opr. Graficzne — Wanda Roszkowska
Zdjęcia — Leopold Dzikowski

Szefowie klanów:
Amstrad — Jonasz Mayer
Atari — Wojciech Zientara
Commodore — Klaudiusz Dybowski
Co jest grane — Lukasz Czekajewski
IBM — Marcin Borkowski
Micro Magazyn — Janusz Jarmoch
Po dzwonku — Tadeusz B. Mańk
Spectrum — Maciej Pietraś

Stali współpracownicy:
Grzegorz Bujanowski
Marek Czarkowski
Robert Magdziak
Waldemar Nowak
Marcin Przasnyski
Maria Radzimińska
Marek Sawicki
Piotr Sumara
Stanisław Szczygieł
Anna Uhera-Mlonek
Rafał Wiosna
Stanisław Winiecki
Beata Znamirowska

Bajtek BBS:
SysOp — Rafał Wiosna
Telefon — 635-59-04

Wydawca:
Spółdzielnia „Bajtek"
ul. Wspólna 61
00-687 Warszawa

Skład i druk:
Przedsiębiorstwo Poligraficzno-Wydawnicze „Gryf" Sp. Akc. Ciechanów
Fotoskład — Grzegorz Simiński
Montaż — Grażyna Ostaszewska
Korekta — Maria Krajewska
Teresa Rutkowska
Nakład 102 tys. egz. Zam. 39601

Redakcja nie odpowiada za treść ogłoszeń.
Nie zwracamy materiałów niezamówionych, za wyjątkiem nośników.

Celem ułatwienia zainteresowanym kontaktów z zespołami poszczególnych klanów, stworzyliśmy system dyżurów. Prosimy dzwonić w określonych dniach i godzinach.
Amstrad — środa 10–12
Atari (Moje Atari) — czwartek 12–16
Commodore (C-64, Amiga) — środa 16–18
Co jest gane (Top Secret) — środa 12–16
Po dzwonku (Edukacja) — wtorek 12–16
IBM — czwartek 15–18
Spectrum — czwartek 14–16

Magic Clip



W krótkiej relacji z poznańskich targów InfoSystem pisaliśmy między innymi o pojawieniu się różnych akcesoriów, wspomagających i ułatwiających wykorzystanie komputera. Jednym z nich jest Magic Clip — mocowany do monitora uchwyt do kartek papieru. Od producenta Magic Clip-a, firmy Polplast z Łomianek, otrzymaliśmy kilka sztuk uchwytu do używania w redakcji.

Jak sprawował się Magic Clip w redakcji? Bardzo dobrze. Przy naszych komputerach przyklejonych było kilka sztuk uchwytu, i żaden nie sprawiał nam kłopotów — wszystkie trzymały się mocno monitorów, żaden nie pękł, mimo grubych plików kartek wepchniętych w jego szczęki, a papier trzymał się w ustalonej pozycji niezależnie od tego czy była go jedna cienka stroniczka, czy kilka. Ponieważ czas, jakim dysponowaliśmy był zbyt krótki, byśmy mogli mieć pewność że uchwyty nie odkleją się od monitorów po kilku miesiącach, pytałem o opinię wszystkich napotkanych użytkowników Magic Clip-a. Poza jednym przypadkiem nikt nie narzekał — w Apexim-ie Magic Clip odkleił się od monitora po jakimś czasie, jednak niewykluczone, że stało się tak dlatego, że powierzchnia do której został przyklejony była brudna. Żeby tego uniknąć, wystarczy umyć dokładnie monitor przed przymocowaniem uchwytu (najlepiej przetrzeć powierzchnię plastiku watą zwilżoną spirytusem).

Jako znany malkontent, muszę wrzucić do ogródka Polplastu jeden kamyczek. Przykleiłem Magic Clip do pleców ekranu laptopa, z którego korzystam, i okazało się, że budowa uchwytu nie przewiduje jego zablokowania w pozycji całkowicie rozgiętej. Przeznaczone do tego celu ząbki blokują uchwyt tylko wtedy, gdy jest on zgięty pod kątem bliskim prostemu. Wprawdzie nie uniemożliwia to korzystania z Magic Clipa-a przymocowanego do laptopa, ale też nie ułatwia pracy — przy poruszeniu komputera uchwyt przekrzywia się pod wpływem obciążenia i kartka ginie z pola widzenia.

Marcin Borkowski

Producent Magic Clip-a:
Polplast
ul. Kościuszki 5
05-092 Łomianki
tel. 6417105

SuperDisk



Dyskietka towarzyszy wielu wydawanym na Zachodzie magazynom komputerowym. Przeważnie znajdują się na niej programy, których listingi zostały opublikowane w piśmie. Ma to oszczędzić czytelnikowi kłopotów związanych z samodzielnym wklepywaniem treści programu.

Niektóre redakcje nie drukują listingów, a mimo to pismo sprzedawane jest z dyskietką. Do takich czasopism zalicza się brytyjski „PC Plus". W cenę magazynu, która wynosi 2,60 funta, wliczona jest dyskietka SuperDisk. Zawiera ona prawie 350 KB całkiem niezłego oprogramowania dla użytkowników komputerów osobistych PC. Można tam znaleźć grę, ciekawy program użytkowy lub interesujące procedury. Dotychczas ukazało się ponad 50 wydań dyskietki.

Spora część materiałów wydawanych na SuperDisk jest dziełem czytelników PC Plus. Za każdy program można otrzymać od 20 do 750 funtów. Najwyższa kwota jest wypłacana oczywiście za duże, najlepiej opracowane programy, ale i krótkie procedury są nagradzane kilkudziesięcioma funtami. Wskazówki dla przyszłych autorów znajdują się w każdym numerze pisma. Nadsyłane programy nie mogą naruszać praw autorskich innych osób i nie mogą być publikowane w innych pismach. Programista musi również przewidzieć możliwość uruchomienia programu na sprzęcie o różnych konfiguracjach i dołączyć dobry opis w zbiorze z rozszerzeniem DOC.

Drugą grupę programów publikowanych na SuperDisk stanowi tak zwane shareware. Użytkownik może korzystać z niego bezpłatnie do 30 dni. Instrukcja obsługi i opis programu są zamieszczone także w czasopiśmie. Po upływie trzydziestu dni należy się zarejestrować u dystrybutora i wnieść pewną opłatę.

Dyskietka zawiera także programy demonstracyjne. Działają zupełnie normalnie, ale są nieco zubożone w stosunku do wersji handlowej. Ograniczenie polega często na zablokowaniu funkcji zapisu wyników na dyskietce. Właściciel dyskietki demonstracyjnej może ocenić przydatność programu użytkowego i trochę pograć w grę. Niestety, pracę lub zabawę trzeba zaczynać zawsze od początku. Pełną wersję programu i prawo do stałego użytkowania uzyskuje się oczywiście po wpłaceniu określonej kwoty na konto producenta.

(J)

PIRACI w sądzie

Brytyjska firma Marconi Instruments została oskarżona o rzekomo nielegalne kopiowanie oprogramowania. Jest to pierwszy w Wielkiej Brytanii przypadek wytoczenia procesu instytucji, która kopiowała oprogramowanie tylko do użytku wewnętrznego, a nie na sprzedaż. Akcja została podjęta przez czterech potentatów software'owych: Ashton-Tate, Lotus Development, Microsoft i WordPerfect. Firmy te są członkami związku Business Software Alliance (BSA) i organizacji Federation Against Software Theft (FAST), która zajmuje się zwalczaniem piractwa.

Ze względów prawnych nie ujawniono, które pakiety oprogramowania zostały skopiowane i ile kopii użytkowano. Rzecznik BSA oświadczył, że poszkodowani producenci oprogramowania nie wkroczyliby na drogę postępowania sądowego, jeśli w rachubę wchodziłaby niewielka liczba kopii.

Rewizji w pomieszczeniach Marconi Instruments dokonano 22 listopada 1990 r. po uzyskaniu zgody sądu na inspekcję komputerów osobistych PC znajdujących się w siedzibie firmy.

Według BSA przypadek ten jest pierwszy z całej serii akcji podjętych w Europie. W Wielkiej Brytanii BSA skontaktowała się z czternastoma innymi przedsiębiorstwami i ostrzegła je przed konsekwencjami naruszania praw autorskich. Zgodnie z nakazem wydanym przez sądy francuskie przeprowadzono w grudniu ubiegłego roku dwie dalsze inspekcje. W jednym przypadku skierowano sprawę do sądu.

(J)

Na podst. PC Plus

Bezprzewodowa mysz

Firma Logitech skonstruowała bezprzewodową odmianę myszy MouseMan. Urządzenie jest nieco większe niż jego konwencjonalny odpowiednik. Bezprzewodowa mysz odziedziczyła po swojej poprzedniczce charakterystyczny kształt oraz ergonomiczną konstrukcję.

Dotychczas podejmowano próby skonstruowania bezprzewodowych myszy, które przekazywały sygnały za pomocą wiązki promieni podczerwonych. Mysz Logitech komunikuje się z komputerem PC drogą radiową za pośrednictwem sygnałów o niskiej częstotliwości. Odbiór sygnałów umożliwia niewielki odbiornik, który przyłącza się do PC poprzez interfejs szeregowy. Maksymalna odległość odbiornika od myszy wynosi 2 metry. Częstotliwość sygnałów reguluje się w zakresie od 100 do 150 KHz. Użytkownik może wybrać jeden z ośmiu kanałów w tym pasmie. Pozwala to uniknąć konfliktów wynikających z pracy dwóch myszy w niewielkiej odległości od siebie. Zmiana kanału jest bardzo prosta. Służy do tego lewy przycisk urządzenia. Dioda LED umieszczona na grzbiecie myszy mruga od 1 do 8 razy w zależności od numeru kanału.

Myszy komunikujące się z komputerem za pomocą promieni podczerwonych miały jeden istotny mankament. Użytkownik musiał stale uważać, by wiązka promieni była wycelowana w odbiornik i żeby na jej drodze nie znajdowały się żadne przeszkody. Każde odchylenie, nawet o kilka stopni od właściwego kierunku, natychmiast przerywało komunikację. Mysz radiowa nie ma tych wad.

Stosowanie promieni podczerwonych wiązało się również ze stosunkowo dużym zużyciem energii. Radiowa mysz Logitech jest zasilana z niewielkiej baterii litowej, która wystarcza na rok pracy. Osiągnięcie tak długiej trwałości baterii było możliwe dzięki zastosowaniu prostego układu elektronicznego, który przerywa komunikację z komputerem, gdy mysz jest nie wykorzystywana.

(J)





TEXTON wielojęzyczny słownik

Izraelska firma S.E.M. Ltd oferuje wielojęzyczny słownik komputerowy o wymiennych pakietach dla różnych języków. Pakiet hebrajsko-angielski zawiera około 250 tys. słów, obejmując również zwroty charakterystyczne dla amerykańskiego slangu. Czterowierszowy wyświetlacz ciekłokrystaliczny oraz niewielka klawiatura o standardowym układzie zapewniają wygodną pracę.

(JM)

Niezawodny komputer

Od sprawności komputerów zależy dziś wiele dziedzin życia. Awaria maszyn liczących zainstalowanych w dużym przedsiębiorstwie, na przykład w banku, powoduje poważne straty finansowe. Uszkodzenie systemów komputerowych pracujących w towarzystwach lotniczych może mieć katastrofalne następstwa dla życia ludzkiego. Dlatego coraz więcej użytkowników przywiązuje duże znaczenie do odpowiednich zabezpieczeń minimalizujących ryzyko unieruchomienia komputera. W najbardziej niezawodnych systemach wszystkie ważne obwody są dublowane. Dwa niezależne komputery pracują równolegle. W przypadku awarii jednego z nich drugi może natychmiast kontynuować realizację zadania. Systemy te są bardziej odporne na katastrofy, sabotaż i skutki akcji terrorystycznych.

Tak skomplikowane urządzenia mają swoją cenę. Niegdyś na zakup systemów podobnej klasy mogły sobie pozwolić tylko najbogatsze przedsiębiorstwa. Ogólny spadek cen w branży komputerowej przyczynił się do znacznego rozpowszechnienia tych urządzeń.

Instytuty naukowe prognozują systematyczny wzrost zapotrzebowania na nieawaryjne komputery. W Europie w 1989 roku rynek niezawodnych systemów wynosił 646 milionów dolarów. Do 1995 roku wartość sprzedaży powinna wzrosnąć do 2,1 miliarda dolarów. Specjaliści przewidują największy popyt w Wielkiej Brytanii (555 mln $), RFN (538 mln $) i Francji (371 mln $). Największym odbiorcą bezpiecznych systemów komputerowych będą banki i inne instytucje finansowe, które w sumie wydadzą na ten cel ponad 843 mln $.

na podst. CHIP

(J)

CAH

Wśród nowych produktów bogatego światowego rynku programów użytkowych pojawił się ostatnio groźny konkurent dla znanych pakietów CAD i CAM. Program CAH (Computer Aided Housekeeping), proponowany przez spółkę Wives Ltd., składa się z kilku formalnie niezależnych części, zintegrowanych w jeden pakiet. W jego skład wchodzą: program budżetowy (potrafiący za pomocą modemu skontaktować się z siecią bankową i sprawdzić stan konta), program do ustalania jadłospisu (potrawy, dopasowane za pomocą pliku konfiguracyjnego do gustu rodziny, można wybierać z rozwijanych menu), system zarządzania zapasami domowymi (połączony z czytnikiem kodów prążkowych umieszczonym w lodówce) oraz system ekspercki pomagający leczyć najprostsze choroby sposobami domowymi. Całość wymaga dowolnego komputera z MS-DOS-em 2.0 lub nowszym, jednej stacji dyskietek i zaledwie 353,5 KB RAM-u.

(mb)

KONIEC Z SZAROŚCIĄ

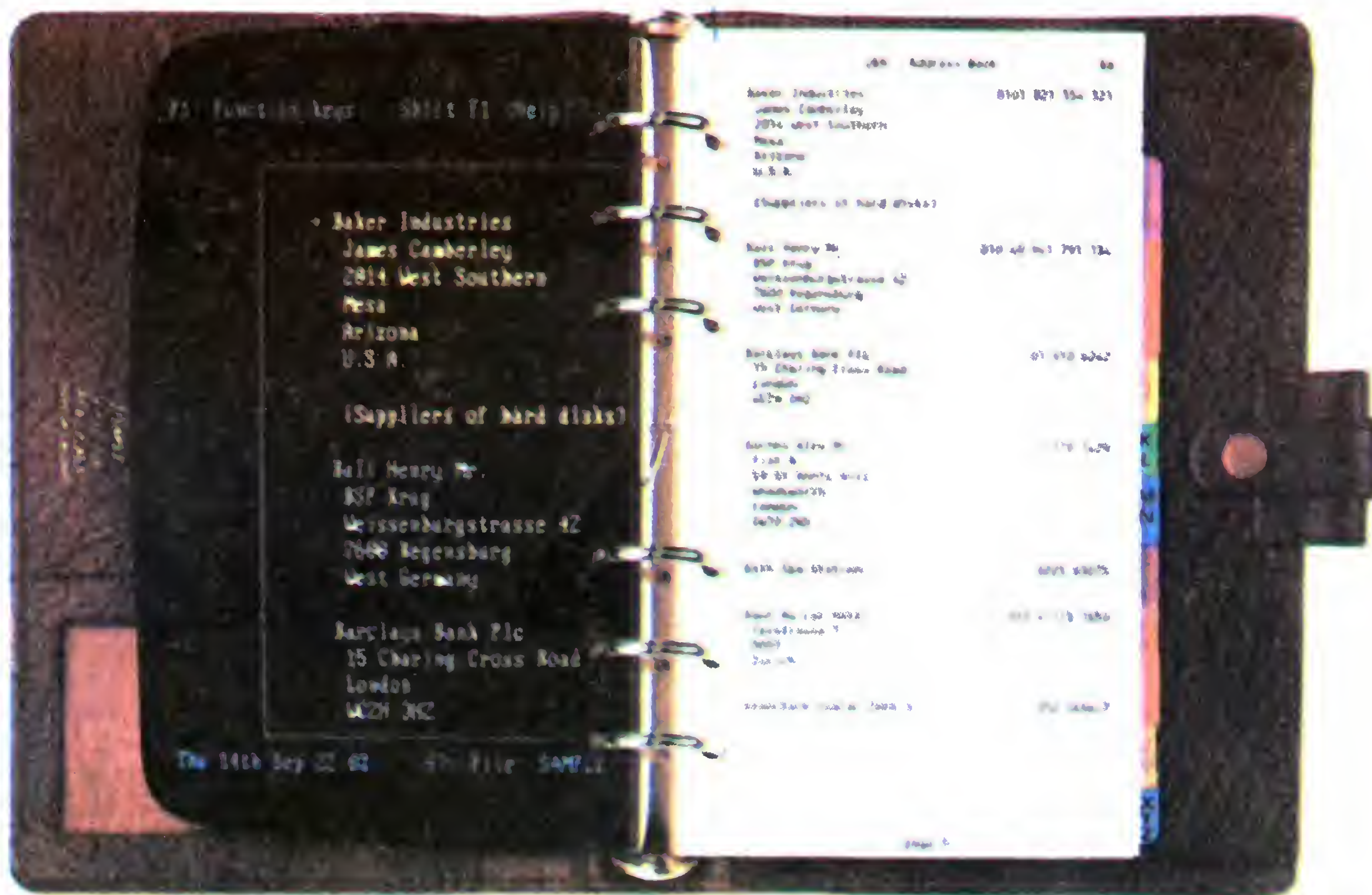
Kolorystyka komputerów osobistych sprawia bardzo przygnębiające wrażenie. Na wystawach widzi się przede wszystkim różne odcienie szarości i beżu, a tylko niekiedy elegancką czerń lub brąz. Te nieciekawe barwy mają zwiększyć komfort pracy i są, zdaniem specjalistów od ergonomii, bezpieczniejsze dla zdrowia. Szara obudowa tłumi refleksy światła nawet do 50 % i uspokaja wzrok.

Tymczasem użytkownicy nie zawsze podzielają opinie fachowców. Firma Commodore wyprodukowała informacyjną serię Amigi 500, ozdobionej kolorowymi kleksami i upstrzonej tygrysimi cętkami. Komputery, pomimo wyższej ceny, sprzedano błyskawicznie.

Badania przeprowadzone wśród użytkowników PC wykazały, że wielu z nich ma już dość szarości. Komputer powinien mieć swój indywidualny styl, o którym decyduje również odpowiednio dobrana barwa. Kolor i kształt obudowy musi właściwie harmonizować z otoczeniem, a także odpowiadać temperamentowi i upodobaniom użytkownika. Zdaniem sporej grupy osób komputer musi pasować do wystroju wnętrza i do innych mebli stojących w pomieszczeniu. Tradycyjna kolorystyka nie zawsze spełnia te warunki.

(J)





BAZY DANYCH

Już starożytni Rzymianie mieli bazy danych! Stwierdzenie to nie jest wcale bezsensowne, choć może się takim wydawać. Jest to wynik wieloletnich badań, całkowicie oparty na sprawdzonych danych z mojej bazy.

Chyba każdy z nas ma jakiś notes lub zeszyt z numerami telefonów i adresami znajomych. Gdy taki notes otworzyć, zobaczymy w nim mniej więcej taki zapis:

Jan Kowalski
tel. 85-32-41
02-775 Warszawa
ul. Majonezowa 3 m 8
kolega z obozu harcerskiego

Podobne zapisy zawiera książka telefoniczna — tyle, że jest ona znacznie grubsza, gdyż zawiera tysiące nazwisk i numerów. Mimo to, jest ona właściwie tym samym co nasz mały notesik. Podobnie np. ewidencja ludności (spis mieszkańców), katalog książek w bibliotece, ewidencja samochodów, lista płac w przedsiębiorstwie — wszystko to są dane, podobne do danych w naszym notesie (mimo, że różnią się treścią i objętością). Są to dane zebrane w pewne jednostki (np. dane o panu Kowalskim, jego adresie i numerze telefonu), w pewien sposób zebrane razem i ułożone w pewnej kolejności (w przypadku notesu jest to kolejność alfabetyczna nazwisk naszych znajomych). Czyli — są to wszystko BAZY DANYCH!

Zaraz odkopiemy tę teczkę...

Każdy, kto korzystał z Czytelni Głównej Biblioteki Miasta Stołecznego Warszawy przy ulicy Koszykowej spotkał się zapewne z problemem podobnym do mojego: Jak znaleźć — wśród ponad 350 tysięcy — kartę z danymi o potrzebnej akurat książce. Zabiera to zwykle około pół godziny — i to tylko dzięki temu, że katalogi są podzielone według tematów i autorów (katalog tematyczny i alfabetyczny). Co jednak zrobić, gdy mamy kilkaset tysięcy kart, których nie można w ten sposób podzielić — np. spis telefonów...

Trzeba tracić cenny czas na grzebanie w stertach papierów, wdychać kurz i tracić wzrok czytając pożółkłe akta. Podobnie gdy chcemy uaktualnić dane — np. jeśli nasz pan Kowalski przeniósł się do sąsiedniego bloku i trzeba zmienić adres zostawiając numer telefonu. Chyba z tego powodu książka telefoniczna jest wydawana raz na kilka lat (i nieaktualna już w momencie przekazania do druku...).

A jeśli komuś przyjdzie do głowy zrobić spis blondynów, których numer telefonu kończy się na 7, dodatkowo posiadających samochody wyprodukowane po roku 1987? — Proste, musi przejrzeć trzy bazy danych: ewidencję ludności, ewidencję samochodów i spis telefonów. KOSZMAR!

Inteligentne narzędzie

Ratunkiem jest użycie inteligentnego narzędzia — komputera. Oczywiście, komputer może zrobić z danymi co tylko zechcemy, trzeba je tylko przedstawić w odpowiedniej formie i wydać właściwy rozkaz.

Całą bazę danych (np. notes) trzeba podzielić na pojedyncze jednostki, odnoszące się do jednej osoby (jednego przedmiotu itp). Taką część składową bazy danych nazywamy **rekordem**. W przypadku notesu, rekordem będzie wszystko co zapisaliśmy na temat pana Kowalskiego — jego nazwisko, adres, numer telefonu i ewentualnie jakieś uwagi. Część rekordu — np. zawierającą nazwisko — nazywamy **polem rekordu**. Rekord opisujący pana Kowalskiego można podzielić na następujące pola: *imię, nazwisko, kod pocztowy, miasto, ulica, numer domu, numer mieszkania, telefon i komentarz.* W ten sposób mamy już rekord i pola — czyli części składowe bazy danych. Sama baza danych jest zbiorem rekordów — a każdy rekord zbiorem pól.

Pole polu nie równe

Pola bywają różne: znakowane i numeryczne, długie i krótkie, pełne i puste. Pola **znakowe** służą do przechowywania tekstu, np. nazwiska, natomiast pola **numeryczne** (czyli liczbowe) — do przechowywania liczb (np. numeru mieszkania). Istnieją także inne rodzaje pól, ale są one rzadziej stosowane.

Długość pola określa ilość informacji, jaką możemy w tym polu umieścić. Pole znakowe o długości 20 pozwala zapisać dwadzieścia znaków (zwykle tyle wystarcza na nazwisko). Pole numeryczne o tej samej długości pozwoliłoby na zapis dwudziestocyfrowej liczby. W przypadku pól numerycznych występuje jeszcze jeden parametr: ilość miejsc po przecinku.

Pole może być puste jeśli nie umieścimy w nim żadnej danej. Pola numeryczne zachowują się wtedy tak, jakby zawierały zero.

Z papieru do pamięci

Wróćmy do naszego notesu — jest on zbiorem rekordów. Jeśli przeniesiemy taki zestaw rekordów do komputera, utworzą one plik. Ten plik będzie można przeglądać tak jak notes — ale inne operacje będą znacznie prostsze.

Wyobraźmy sobie teraz, że podczas wakacji



1 — Pod tym nagłówkiem podawane są ilosci miejsc po przecinku (tylko pola numeryczne)
2 — Nazwa pliku z bazą danych
3 — Ilość rekordów danych w bazie
4 — Data ostatniej modyfikacji danych

```
Structure for database: C:test.dbf       [2]
Number of data records:          1       [3]
Date of last update    : 07/01/91        [4]
Field  Field Name   Type       Width   Dec
    1  NAZWISKO     Character     40    [1]
    2  IMIE         Character     15
    3  KOD_POCZT    Character      8
    4  MIASTO       Character     30
    5  ULICA        Character     30
    6  NR_DOMU      Numeric        5
    7  NR_MIESZK    Numeric        5
    8  MA_TELEFON   Logical        1
    9  NR_TELEFON   Character     10
   10  DATA_WPISU   Date           8
   11  UWAGI        Memo          10
** Total **                      163
```

Długość pola **Nazwa pola** **Typ pola** **Łączna długość rekordu**

poznaliśmy kilkanaście osób, które chcemy teraz umieścić w notesie. I tu kłopot — już po wpisaniu połowy skończyło się miejsce na odpowiednich stronach. I co teraz? Wpisać Asię Kwiatkowską pod literą „L"? Potem mogą być kłopoty z odszukaniem jej adresu. Założyć drugi notes? Też niezbyt wygodne.

W przypadku komputerowej bazy danych nie ma takiego problemu. Po prostu dodajemy nowy rekord i **sortujemy** (układamy rekordy w kolejności — a właściwie nie my, tylko program obsługujący bazę danych) plik zawierający bazę danych. Znowu mamy wszystkich w kolejności alfabetycznej.

A zmiana adresu lub telefonu? Zamiast kreślić i dopisywać z boku maczkiem po prostu zmieniamy zawartość odpowiedniego pola w konkretnym rekordzie i tyle.

Możliwość zwykłego **przeglądania** jest oczywista. Korzystając z komputera możemy także przeglądać jedynie wybrane, za pomocą **klucza**, rekordy. Na przykład możemy wybrać sobie tylko te rekordy, które odnoszą się do osób mieszkających w Warszawie. Z tak wybranych rekordów można bez problemu zbudować drugą bazę danych. Oczywiście klucz można wybrać zupełnie dowolnie, można też użyć kilku kluczy naraz, np. osoby z Warszawy, mieszkające na ulicy Marszałkowskiej lub Wspólnej — czyli wybór według pól *miasto* i *ulica*.

W przypadku „poważniejszych" baz danych, ważne są także **raporty**. Większość programów obsługi baz danych (np. dBase, FoxBase, Masterfile i inne) ma możliwość automatycznego — oczywiście na podstawie zadanych parametrów — tworzenia raportów. Raport może zawierać np. informacje ilu z naszych znajomych posiada telefon, listę nazwisk z numerami. W przypadku np. ewidencji ludności, raport może zawierać dane o mieszkańcach danej ulicy, średnią wieku, płeć itp. Jak to zwykle w przypadku komputera — zrobi on to, co rozkażemy.

Wszystko co każemy — ale trzeba wydać odpowiedni rozkaz, czyli napisać program do obsługi naszej bazy danych. Informatyk czy programista zrobi to bez specjalnych problemów. Typowy użytkownik bazy danych, czyli urzędnik, magazynier lub dyrektor zwykle nie zna się na programowaniu w wystarczającym stopniu. Nic więc dziwnego, że wkrótce po rozpowszechnieniu się komputerów osobistych powstały programy, pozwalające każdemu założyć i obsługiwać własną bazę danych. Jednym z pierwszych i chyba najbardziej popularnych jest **dBase**.

To długa historia...

Tak, dBase ma długą historię. Zaczęło się to w czasach, gdy niepodzielnie panował CP/M — pierwszy system operacyjny „z prawdziwego zdarzenia" na komputery osobiste. Były to czasy, gdy o komputerowych bazach danych mało kto słyszał — za wyjątkiem operatorów komputerów *mainframe* — czyli dużych, stacjonarnych systemów, w których jeden komputer zajmował całą halę i wymagał klimatyzacji. Odpłacał za to dużą (jak na owe czasy) mocą obliczeniową i możliwością przechowywania dużych ilości danych (głównie na taśmach magnetycznych).

dBase otworzył świat baz danych dla użytkowników „małych" komputerów — komputerów osobistych. W przypadku komputerów ośmiobitowych, rozwój zatrzymał się na wersji II, mogącej pracować na każdym komputerze wyposażonym w system CP/M — można tu wymienić Amstrada, Commodore C-128, Spectravideo (SVI), Juniora i poczciwe ZX Spectrum ze stacją FDD 3000.

Gdy komputery osobiste rosły, obrastając megahercami i kilobajtami, wraz z nimi wzrastał dBase, do którego wkrótce dołączyli konkurenci. dBase stał się jednak standardem. Obecnie program, który nie ma możliwości co najmniej najnowszej wersji dBase nie ma na rynku oprogramowania dla komputerów szesnastobitowych większych szans. Konkurencja prześciga się w coraz to wymyślniejszych rozwiązaniach — na przykład najnowsza wersja opisywanego przez Marcina Borkowskiego programu FoxBase ma możliwość przeszukania bazy składającej się z kilku milionów rekordów w ciągu kilku sekund! Program ten powstawał jako „szybszy dBase" — co udało się osiągnąć. Ciekawe, czym odpowie firma Ashton-Tate?

Mimo że nie powstają już nowe wersje dBase „pod" CP/M, dane nie są stracone. Korzystając z typowych możliwości dBase można je łatwo przenieść na IBM PC i przerobić na *format nowszej wersji dBase (np. dBase IV)* — jeśli ktoś ma taki problem, to Stanisław Szczygieł właśnie go rozwiązał. Wystarczy przeczytać i zastosować podaną metodę.

dBase — BASIC baz danych

Program dBase (firmy Ashton-Tate) był pierwszym w pełni profesjonalnym i popularnym systemem obsługi baz danych. Jego podstawowe możliwości to typowe operacje, takie jak tworzenie, przeglądanie i edycja baz danych. Wygoda dBase polega na tym, że do prostych działań nie są potrzebne żadne specjalne kwalifikacje — wystarczy poznać kilka podstawowych komend. Znając komendy podane w ramce „Podstawowe komendy dBase" można stworzyć i używać praktycznie dowolną bazę danych.

Nowy język

dBase jest nie tylko systemem obsługi baz danych — jest także interpreterem specjalnego języka obsługi baz danych. Oznacza to, że można w języku dBase napisać program całkowicie sterowany wybieraniem opcji z menu (ang. menu driven), który wykona wybraną operację nie wymagając wchodzenia w szczegóły. Jest to wielka zaleta dla użytkowników nie znających się na programowaniu — jeśli nie chcą (lub nie mają czasu) uczyć się podstaw programowania, mogą oni zamówić lub kupić gotowy program, który korzystając z możliwości dBase wykona „brudną robotę" po naciśnięciu jednego klawisza.

Natomiast osoby piszące programy szczególnie cenią interakcyjność dBase, podobną

PODSTAWOWE KOMENDY dBase

Komendy podane w tej ramce to „podstawy podstaw", jednak w zupełności wystarczające do prostej obsługi bazy danych. Nie jest to jednak opis pełny — trzeba znaleźć informacje o składni podanych komend. Podany tu opis jest jedynie *„orientacyjny"*. Dokładniejszy opis komend można znaleźć w instrukcji lub w jednej z kilku książek o dBase.

CREATE — tworzenie bazy danych, definiowanie struktury (nazw, typów i długości pól). dBase pozwala na używanie pól numerycznych (*numeric*, skrót N), tekstowych (czyli znakowych: *character*, skrót C, logicznych (logical, skrót L), zawierających daty (date, skrót D) oraz notatnikowych (memo, skrót M, nie wszystkie wersje).

USE — otwarcie bazy danych (pliku danych). Umożliwia operacje na bazie danych zapisanej na dysku.

EDIT — edycja danych, na ekranie wyświetlany jest wybrany rekord, w którym można zmieniać zawartość pól.

BROWSE — przeglądanie/edycja danych w bazie.

APPEND — dopisywanie danych. Dane dodawane są na końcu pliku bazy danych.

DELETE — kasowanie rekordów. Rekordy skasowane tą komendą nie są niszczone, lecz oznaczane jako skasowane (przy edycji i przeglądaniu pojawia się gwiazdka). Fizyczne skasowanie rekordów zaznaczonych przez DELETE następuje po komendzie PACK.

GOTO — służy do wybierania konkretnego, określonego numerem, rekordu.

LOCATE — wyszukuje rekord na podstawie kryteriów odnoszących się do zawartości jednego (lub kilku) pola.

DISPLAY — służy do wyświetlania części (lub całości) bazy, możliwe jest podanie klucza, według którego wybierane są rekordy do wyświetlania.

DISPLAY STRUCTURE — wyświetla strukturę rekordu: listę pól wraz z nazwami, typami i długościami oraz podaje długość rekordu w bajtach.

SORT — sortowanie bazy danych według podanego klucza. Jest to dosyć długa operacja, założenie indeksów poważnie ją skraca.

INDEX — utworzenie indeksu według podanego klucza. Założenie indeksu przyśpiesza sortowanie i przeszukiwanie bazy danych.

FIND — szukanie podanej danej w bazie.

REPORT — tworzenie raportu, zwykle według przygotowanego formatu.

CHANGE — edycja danych spoza bazy, na przykład CHANGE STRUCTURE — edycja struktury rekordu, CHANGE COMMAND — edycja programu.

do starego, dobrego BASIC-a: tak jak w BASIC-u po wpisaniu „PRINT 2×2” otrzymamy od razu wynik, tak wpisując pojedyncze komendy dBase otrzymujemy wyniki natychmiast po naciśnięciu ENTER.

Oczywiście, ma to pewne ograniczenia — nikt przecież nie będzie wykonywał skomplikowanych obliczeń (np. obliczenia całek oznaczonych metodami numerycznymi) wpisując po tysiąc razy kilka poleceń — każdy rozsądny człowiek napisze program według właściwego dla danej operacji algorytmu, wyda komendę RUN i poczeka chwilę na wyniki (w przypadku obliczania całek, może w międzyczasie iść na kawę). Podobnie mają się sprawy w przypadku dBase.

Znając podstawowe komendy dBase, możemy obsługiwać naszą bazę danych bez problemu. Co jednak zrobić w przypadku gdy przygotowaliśmy bazę z danymi dla kogoś kto nie zna się na programowaniu (np. główny księgowy dostaje listę pracowników na dyskietce)? Wyjściem jest napisanie programu, który nie będzie wymagał wiedzy programistycznej od osoby korzystającej z bazy danych. Taki program powinien wyświetlić menu z kilkoma opcjami, wybieranymi naciśnięciami pojedynczych klawiszy. Każda opcja wybiera konkretną operację, np. przeglądanie, edycję, sortowanie, wybór według klucza lub zakończenie pracy programu. Po napisaniu takiego programu, jedynym co musimy wytłumaczyć użytkownikowi to sposób jego uruchomienia — a jest on bardzo prosty.

Dlaczego by nie napisać programu obsługi bazy danych w innym języku, w Pascalu, C czy BASIC-u? Można tak zrobić, lecz jest to trudne. Powodem jest to, że dBase jest językiem **specjalizowanym**, podczas gdy trzy wymienione poprzednio są językami do ogólnego zastosowania. Język dBase wyposażony jest w komendy przystosowane specjalnie do obsługi baz danych, podczas gdy Pascal czy BASIC komend takich nie mają. W prostym przypadku dodania jednego pola, program w BASIC-u trzeba by praktycznie pisać od nowa, a co najmniej gruntownie przerobić. dBase nie ma takich problemów — plik z bazą danych jest cały czas normalnie obsługiwany, w najgorszym razie nowy rekord nie będzie dostępny dla użytkownika.

Można zrobić więcej

dBase — i inne profesjonalne programy obsługi baz danych — mają jeszcze inne możliwości. Można przyspieszyć wiele operacji na bazie danych — a szczególnie przeszukiwanie, wybieranie rekordów i sortowanie — przez stworzenie **indeksów**. Indeksy to opis kolejności rekordów dla danego klucza. Jeśli, na przykład, posortujemy bazę danych według nazwisk, a chcemy chwilowo przeglądać ją według adresów, normalnie trzeba by ją posortować według adresów a potem powtórnie według nazwisk. Jeśli stworzymy indeksy dla nazwisk i adresów, nie musimy tego robić — wystarczy wybrać indeks, a baza będzie „udawać”, że jest posortowana w sposób wybrany danym indeksem. Programy podobne do dBase pozwalają stworzyć kilka (a nowsze kilkadziesiąt) indeksów dla jednej bazy! Można więc stworzyć indeks dla każdego pola, a później tylko je przełączać. Zmiany zawartości bazy powodują automatyczne uaktualnienie indeksów — zawsze gdy indeksy są aktywne.

Jeśli chcemy wydrukować ładny raport, można stworzyć gotowy **format raportu**, czyli szablon, który zostanie wypełniony danymi z bazy. Podobnie jak w przypadku indeksów, można stworzyć wiele formatów, zarówno ekranowych (używanych przy wyświetlaniu raportu na ekranie) jak i drukarkowych (do wydruków).

A skąd te dane?

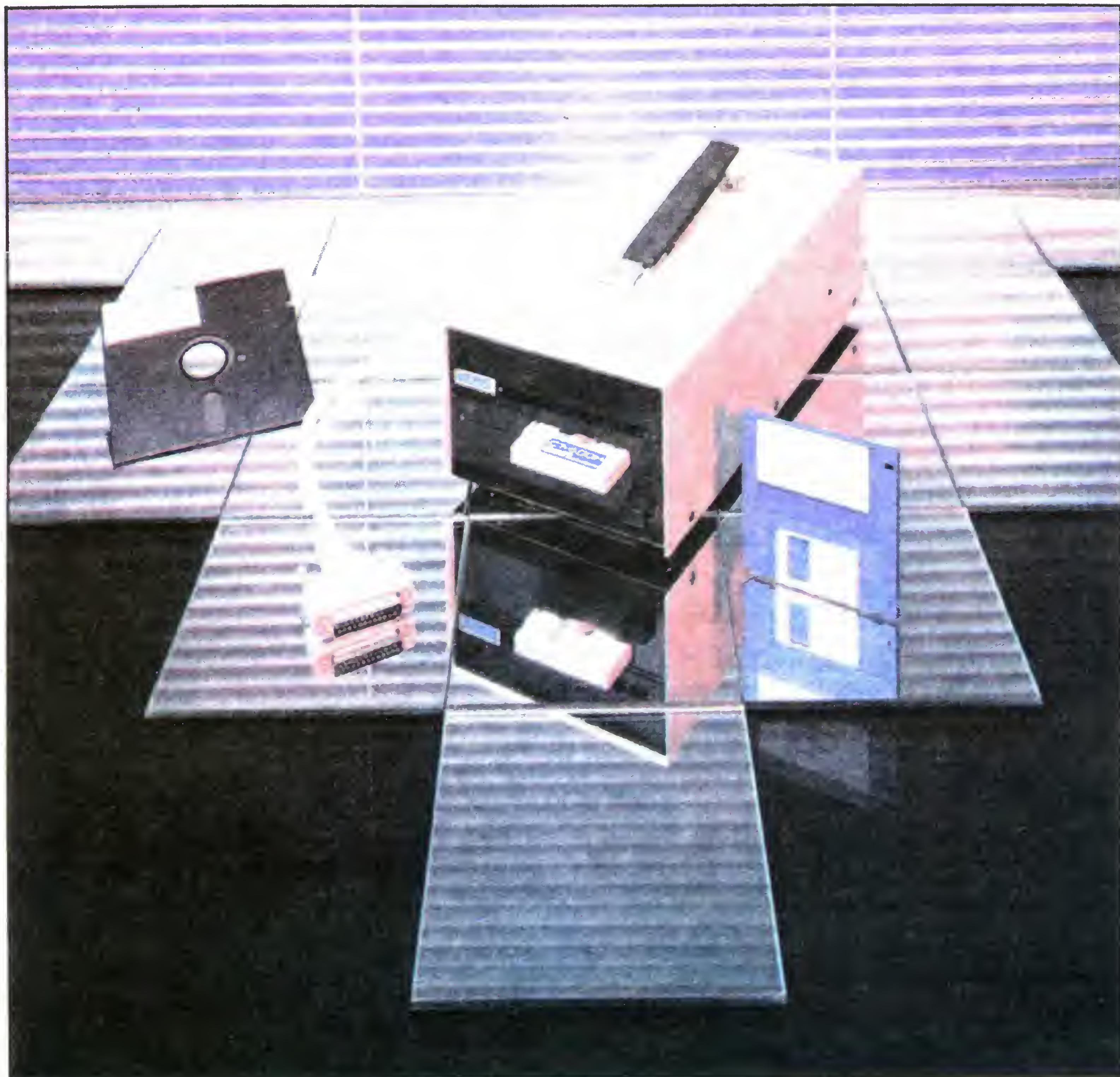
Przenoszenie danych bywa czasami poważnym problemem. Określenie *poważny problem* nabiera pełnego znaczenia w momencie gdy od przeniesienia danych zależy coś więcej niż czyjaś wygoda (co nie znaczy, że nie warto przenosić danych dla ułatwienia sobie korzystania z nich), lub gdy mamy do przeniesienia dużą ilość danych. W przypadku dBase problem ten występuje bardzo rzadko, można bowiem **eksportować** dane, czyli zapisywać je w innym niż normalny formacie. dBase (i większość podobnych programów) ma możliwość eksportowania i **importowania** (operacja odwrotna do eksportowania) w kilku formatach. Tak zapisane dane mogą być później odczytane przez inne programy, na przykład arkusze kalkulacyjne czy edytory tekstu (jeżli jest to format tekstowy). Można także w drugą stronę: przykładowo, większość arkuszy kalkulacyjnych może zapisać dane tak, że można będzie je wprowadzić do bazy danych. Typowym zjawiskiem w świecie baz danych jest czytanie formatu konkurencji, a że dBase jest standardem, wiele programów potrafi czytać wprost ze standardowego pliku dBase oraz bez problemu zapisywać do niego zmienione dane

Mania szybkości

„Skoro mam komputer, to powinienem dostawać wyniki w mgnieniu oka.” — Takie poglądy przejawiała znacząca większość osób, korzystających z baz danych. W przypadku dBase nie zawsze się to udawało, gdyż język dBase jest interpretowany — co powoduje duże straty czasu. Lekarstwem na to okazał się **Clipper** (firmy Nantucket), kompilator języka dBase.

Programy w dBase można więc skompilować, przyspieszając ich działanie nawet kilkadziesiąt razy. Clipper wymaga jedynie niewielkich przeróbek w poddawanych kompilacji programach, czasem nawet udaje się obejść bez żadnych poprawek. Udało się więc osiągnąć znaczną oszczędność czasu (a dla biznesmenów, którzy stanowią znaczny procent użytkowników dBase i podobnych systemów, CZAS TO PIENIĄDZ).

Dodatkową zaletą Clippera jest to, że użytkownik bazy danych nie musi posiadać dBase, gdyż skompilowane programy „radzą sobie same”. Oznacza to oszczędność pieniędzy użytkowników (i straty producenta dBase...). Jeszcze niedawno, wiele polskich firm ofero-



Niektóre funkcje dostępne w języku dBase

INT	—	**jak w BASIC-u, przekształca liczbę rzeczywistą na całkowitą, obcinając część ułamkową.**
STR	—	**zamiana liczby na tekst.**
VAL	—	**działanie odwrotne do STR.**
LEN	—	**podaje długość tekstu danego jako parametr.**
CHR	—	**jak w BASICU-u, przekształca liczbę na znak.**
DATE	—	**podaje datę w postaci tekstu.**
EOF	—	**służy do sprawdzania, czy dotarto już do końca bazy danych (podczas przeszukiwania i innych operacji wykonywanych przez program).**

Pliki danych używane przez dBase

dBase tworząc pliki danych, daje rozszerzenia informujące o zawartości pliku.

DBF	—	**plik bazy danych.**
NDX	—	**plik z indeksem.**
MEM	—	**dodatkowe zmienne.**
CMD	—	**programy w języku dBase.**
FRM	—	**format raportów ekranowych.**
FMT	—	**format raportów drukarkowych.**
DBT	—	**plik zawierający dane z pól notatnikowych (MEMO)**
TXT	—	**dane wyeksportowane w postaci tekstu, przygotowane do przeniesienia do innego programu.**

wało „całkowicie nowe, specjalizowane rozwiązania” systemów baz danych produkowane przy pomocy Clippera. Odbywało się to tak: zespół specjalistów pisał program w dBase, mając możliwość łatwego testowania i poprawiania, by później skompilować go i sprzedawać. Był to swego czasu złoty interes.

Szybkość inaczej

Niektórzy uważali jednak, że chociaż Clipper ma duże możliwości, to jednak czegoś im brakuje. Gdyby tak można było operować na bazach stworzonych przez dBase w jakimś innym języku, na przykład w Pascalu... Okazało się to możliwe. Powstały biblioteki Turbo Database Toolbox, umożliwiające programom w Turbo Pascalu na korzystanie z baz danych w formacie dBase. Oryginalne biblioteki powstały dla wersji 3.0 Turbo Pascala (firmy Borland) na IBM PC. Po głębszym wniknięciu, okazało się możliwe przerobienie tych bibliotek tak, że mogą one pracować na komputerach ośmiobitowych wyposażonych w CP/M (na które dostępne są programy dBase II i Turbo Pascal 3.0). Biblioteki te pozwalają czytać i zapisywać dane do standardowych plików dBase, umożliwiając wielu programistom pisanie programów obsługi baz danych w ich ulubionym języku.

Pisanie w Pascalu pozwala na uzyskanie dużej szybkości (Pascal jest językiem kompilowanym, a Turbo Pascal bardzo skutecznie kompiluje) i jednocześnie eliminuje ograniczenia Clippera, który kompiluje programy przeznaczone do typowych działań na bazach danych. Program obsługi baz danych w Pascalu może zwykle więcej niż skompilowany Clipperem program w dBase, choć oczywiście, jego napisanie wymaga znacznie większej wiedzy.

Podobnie jak sam dBase, biblioteki obsługi baz danych w językach wysokiego poziomu ewoluują, uwzględniając coraz to nowe potrzeby („apetyt rośnie w miarę jedzenia”) użytkowników. Powstają także wersje tych bibliotek w innych językach, w miarę jak wzrasta ich popularność.

Polacy nie gęsi, swoje biblioteki mają! Marcin Borkowski opisuje w tym numerze bibliotekę obsługi baz danych (w formacie dBase) dla Turbo Pascala (wersja 4.0 i nowsze), opracowaną przez polską firmę pana J. Wawro z Jarosławia, pod nazwą Turbo Data Base. Jest to rozbudowany zestaw procedur pozwalający na znacznie więcej niż stary Turbo Database Toolbox.

Pewne wymagania

Każdy może sobie stworzyć własną bazę danych, jednak potrzebny jest do tego odpowiedni komputer. Nie mam tu na myśli żadnego konkretnego komputera lecz pewne wymagania sprzętowe.

Aby zakładanie komputerowej bazy danych miało sens, dane muszą być dostępne natychmiast lub prawie natychmiast, muszą także być bezpieczne — nie życzę nikomu wpisywania kilkudziesięciu kilobajtów po „padnięciu” nośnika, programu lub komputera. Te niewielkie wymagania praktycznie oznaczają konieczność posiadania stacji dysków, bo czekanie za każdym razem, aż załaduje się z taśmy program i dane to zbyt duża strata czasu. Poza tym, kłopoty z ładowaniem z taśmy i straty informacji są większe niż w przypadku używania dysków.

Dobrze również, jeśli komputer jest szybki, oznacza to że szybciej wyszuka interesujące nas dane czy posortuje bazę danych. Nośnik na którym przechowujemy dane powinien mieć dużą pojemność, by nie zabrało miejsca na dysku w połowie wprowadzania danych. Ten warunek spełniają głównie dyski twarde, o pojemności mierzonej w megabajtach.

Jeśli chcemy mieć WŁASNĄ, lokalną bazę danych — musimy mieć odpowiedni komputer w domu lub swoim biurze. W innym przypadku będziemy skazani na nośnik w postaci papieru (notesy, zeszyty, książki) i wszelkie związane z tym niewygody.

To nie żart — można kupić całą, gotową bazę danych. Przykładem może być *Encyclopaedia Britannica*, jedna ze sławniejszych encyklopedii. Można kupić dysk optyczny z odpowiednim interfejsem, podłączyć go do komputera i mieć natychmiastowy dostęp do informacji na wszelkie tematy. Co więcej, encyklopedia ta jest wydawana w formie dyskowej co rok, podczas gdy w formie książki — co kilka lat. Mamy więc informację aktualniejszą niż w książce, a to czasem decyduje o działaniach mających poważne konsekwencje. Podobne bazy danych (również na dyskach optycznych) dotyczą informacji naukowych (chemia, biologia, historia, medycyna itp). Przyłączenie takiej bazy do komputera, to dla naukowca wielkie ułatwienie — zamiast godzinami ślęczeć w bibliotece nad przestarzałymi opracowaniami może, naciskając kilka klawiszy, otrzymać błyskawiczną informację opartą na znacznie nowszych badaniach.

Inną formą sprzedaży baz danych jest sprzedawanie dostępu do baz danych. W wielu krajach, mając modem, można podłączyć się za pośrednictwem linii telefonicznej do najróżniejszych baz danych, zdobywając ciągle uaktualniane informacje. Popularne sieci komputerowe oferują dostęp do ponad 600 różnych baz danych w Europie i USA — miliardy megabajtów świeżej, aktualnej informacji. Za niewielką cenę. Może kiedyś i w Polsce będą powszechnie dostępne zdalne bazy danych... kto wie?

Na razie mamy PESEL — skomputeryzowane dane z Powszechnego Spisu Ludności, dostępne dla jednostek administracji państwowej. Mimo pewnych ograniczeń, system ten wydatnie ułatwia pracę tam, gdzie jest dostępny — łatwiej wyszukać i aktualizować dane. Choć, niestety, wiele jest jeszcze miejsc gdzie wszystkie dane trzymane są na papierze, w ponurych lochach pod nazwą „Archiwum”.

Zakończenie

KTO MA INFORMACJĘ, TEN RZĄDZI. Ta stara maksyma jest w dzisiejszych czasach coraz bardziej prawdziwa. Mając dostęp do informacji można skutecznie działać, nie ryzykując nagłego natknięcia się na nieprzewidziane trudności (mając informację można przewidzieć problemy i zabezpieczyć się). W **społeczeństwie informacyjnym** ten, kto ma informację naprawdę może rządzić. Coraz więcej zależy od szybkiej i dokładnej informacji: gospodarka, polityka, nauka — gromadzenie informacji w bazach danych stało się koniecznością. Kto nie jest dobrze poinformowany, coraz mniej się liczy. Można mu wręcz powiedzieć „Do widzenia, świat się zmienia!”.

Znajomość obsługi komputerowych baz danych, nawet pobieżna, ma obecnie duże znaczenie także dla osób poszukujących pracy — umiejący obsługiwać taką bazę mają szansę na lepsze stanowisko w wielu przedsiębiorstwach, bo coraz częściej segregatory i pliki akt będą zastępowane przez komputer.

A wracając do starożytnych Rzymian... Oni rzeczywiście mieli bazy danych. Aktualizację jednej z nich wspomina nawet Pismo Święte — bo wielki spis ludności to nic innego jak aktualizacja bazy danych o mieszkańcach Imperium. Imperium zaś potrzebowało informacji by zarządzać swoimi prowincjami. Czyli „Już starożytni Rzymianie mieli bazy danych i wiedzieli do czego są one przydatne”.

Michał Szokoło

SZATAŃSKI WIRUS

Pod koniec ubiegłego roku Departament Obrony USA rozpisał konkurs na opracowanie wirusa komputerowego, który mógłby być rozpowszechniany drogą radiową. Nie wiadomo, które z amerykańskich firm software'owych zdecydowały się zainwestować w ten projekt. W każdym razie odezwały się głosy oburzenia wśród uczonych przerażonych wizją „szatańskiego wirusa”. Wszak któż zaręczy, że program będzie atakował tylko „wraże” komputery?

Za oceanem rozpętała się dyskusja nad koniecznością wprowadzenia ograniczeń w nauczaniu studentów metod „produkcji” wirusów komputerowych.

Jeśli nawet uniesieni szlachetnym zapałem jankesi wprowadzą prawo zabraniające podejmowania prac nad tego rodzaju programami. Fred Cohen — jeden z pierwszych teoretyków „wirusologii” — ma godnych następców nad Wisłą.

Otóż Gazeta Stołeczna — dodatek do Gazety Wyborczej — opublikowała 3 lipca 1991 roku tekst pióra niejakiego (smo) pt. Myśli nowoczesnego nauczyciela.

Ów (smo) donosi, że uczniowie klasy informatycznej V Liceum Ogólnokształcącego im. Józefa Poniatowskiego w Warszawie... *piszą poważne programy na konkursy informatyczne, często otrzymują za nie nagrody. Potrafią nawet układać programy wirusowe, a jak wiadomo* — mądrze zauważa (smo) — *stworzenie takiego wirusa, który byłby niewykrywalny przez powszechnie stosowane programy antywirusowe, jest marzeniem każdego* solidnego (podkreślenie własne) *informatyka.*

Tych informacji (jak sądzę) udzielił dziennikarzowi Gazety dyrektor Liceum Mirosław Sosnowski.

Nie jest źle. Jeszcze nie zginęła polska szkoła informatyki skoro tacy nam rosną następcy. Wiadomo, że czym skorupka za młodu nasiąknie tym na starość trąci. Jest szansa, że zdolni licealiści dadzą sobie radę np. z zamówieniem amerykańskiego Departamentu Obrony (głupi Amerykanie sami się wycięli z „przyczyn moralnych”). A my nie takie rzeczy ze szwagrem po pijaku robiliśmy!

Być może za kilka miesięcy dowiemy się, że na zachodniej półkuli pustoszy dyski wirus „O.Take” lub „NIE”.

Życzę zdolnym uczniom (i nauczycielom) V LO TAAKIEGO WIRUSA! A solidnym żurnalistom Gazety równie pasjonujących tematów.

Marek Czarkowski

ps. Wszelkie przypuszczenia, że za „owocną” działalnością uczniów i pedagogów „Poniatówki” ukrywa się autor programu MkS_Vir Marek Sell nie znalazły potwierdzenia.

MODEM TEL-EKO
M-2412 PC

Przed dwoma miesiącami testowaliśmy w „Bajtku" modem M-2412 firmy Tel-Eko. Dzisiaj zajmiemy się jego bratem — modemem M-2412 PC.

OGÓLNIE

M-2412 PC to modem na karcie, tzn. przeznaczony do zamontowania wewnątrz komputera typu PC. Pozwala to na zwolnienie złącza RS-232C, co jest szczególnie ważne dla użytkowników myszy posiadających w komputerze tylko jedno złącze RS. Nie potrzeba też oczywiście zewnętrznego zasilania.

Funkcjonalnie M-2412 PC odpowiada modemowi M-2412, lecz realizuje tylko transmisję asynchroniczną. Oznacza to, że może pracować tylko na zwykłych liniach telefonicznych. Dla zwykłego użytkownika nie ma to w zasadzie znaczenia, tym bardziej w Polsce, gdzie telekomunikacja jest jeszcze w powijakach.

Opisywanie parametrów modemu, trybów pracy, zgodności ze standardami mija się z celem, gdyż komplet takich informacji podał Michał Szokoło w teście modemu M-2412 w „Bajtku" 6/91. Dość przypomnieć, że oba modele są zgodne ze standardami komunikacyjnymi CCITT oraz BELL. Pozwala to na korzystanie zeń praktycznie na całym świecie, homologacje posiada zaś już w pięciu krajach.

BLIŻEJ

Karta modemu wykonana jest bardzo dokładnie i schludnie. Nie ma mowy o niedopasowaniu do złącza na płycie głównej, czy za krótkim slocie. Od strony wykonania modem prezentuje poziom bardzo wysoki.

Komplet stanowi pudełko, zawierające, oprócz karty umieszczonej w foliowej torebce, kabel podłączeniowy do gniazdka telefonicznego oraz instrukcję. Jedyne, czego się można czepiać to opakowanie, wykonane z cienkiej tektury.

RYS HISTORYCZNY

Modem na początku trafił w ręce naszego współpracownika, posiadającego XT. Po trzech dniach skwaszony oddał go twierdząc, że modem nie działa. Po tygodniu dostałem go ja. Modem kontynuował strajk, odmawiając pracy w AT. Za karę przeleżał miesiąc w szufladzie, gdy Naczelny przypomniał sobie o nim. Wstawił go do redakcyjnego AT i — o, wstydzie! — modem zaskoczył.

Nie wiemy, co było powodem takiego zachowania się modemu. Jeśli jednak ten egzemplarz trafiłby w ręce kupującego, nie obyłoby się bez reklamacji.

Od tamtego czasu z modemem nie ma problemów. Pracował w różnych modelach XT oraz AT, często po kilkanaście godzin.

NA FRONCIE

Nie oszczędzałem modemu. Średnio licząc przemielił on ponad 30 MB samych downloadów, a łączny czas pracy szacuję na 200 godzin. Nie licząc kilku zawieszeń nieocenionego programu MTE (emulującego protokoły korekcji błędów, bez czego w RP ani rusz) mam tylko jedną uwagę: przy rozpoczęciu „dialowania" (czyt: wykręcanie numeru), jeśli sygnał centrali nie odezwie się od razu, a po krótkim chrupnięciu, program terminala stwierdza „No DialTone detected by modem".

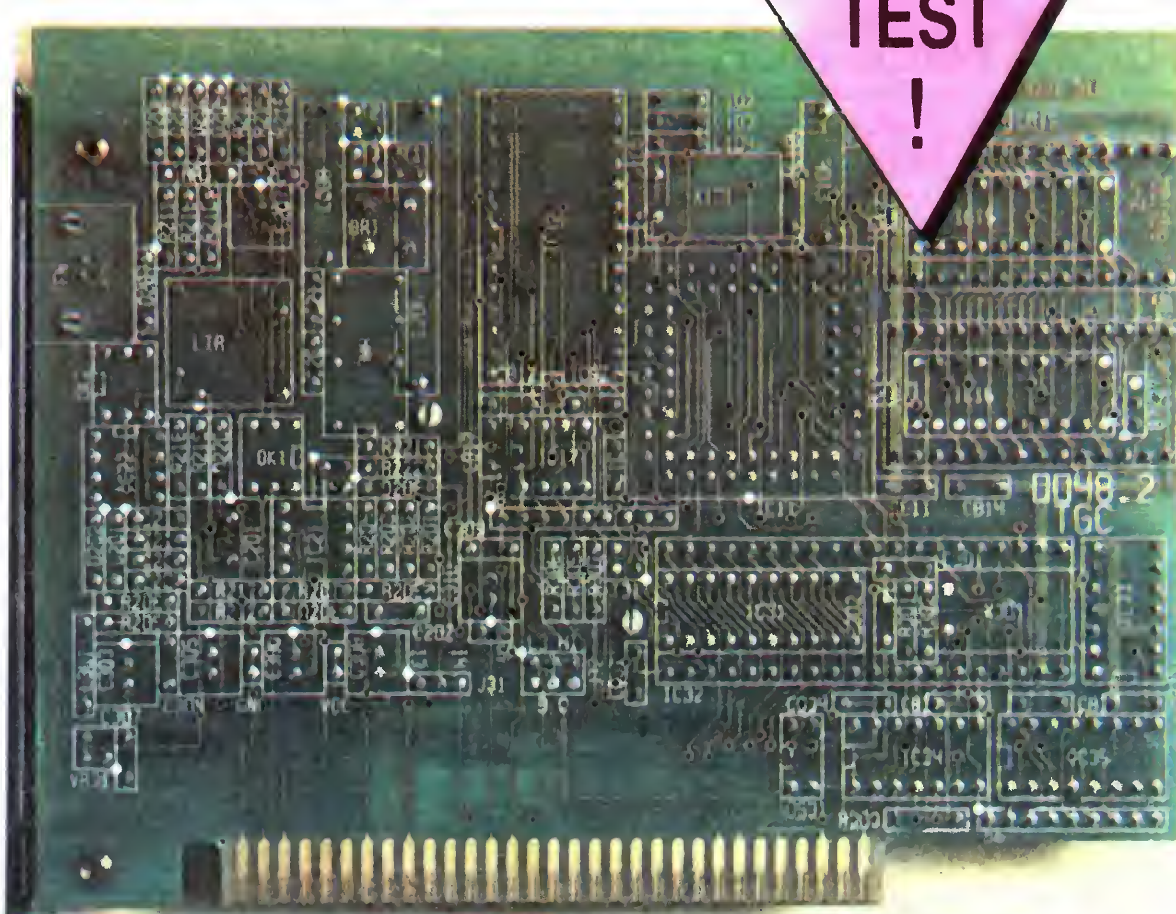
Zamontowany na karcie głośniczek pozwala na wykrycie sygnału zajętości wcześniej, niż sam modem. Praca byłaby wygodniejsza, gdyby głośniczek miał nieco większą moc, gdyż czasem wręcz nie słychać go zza szumu wentylatora.

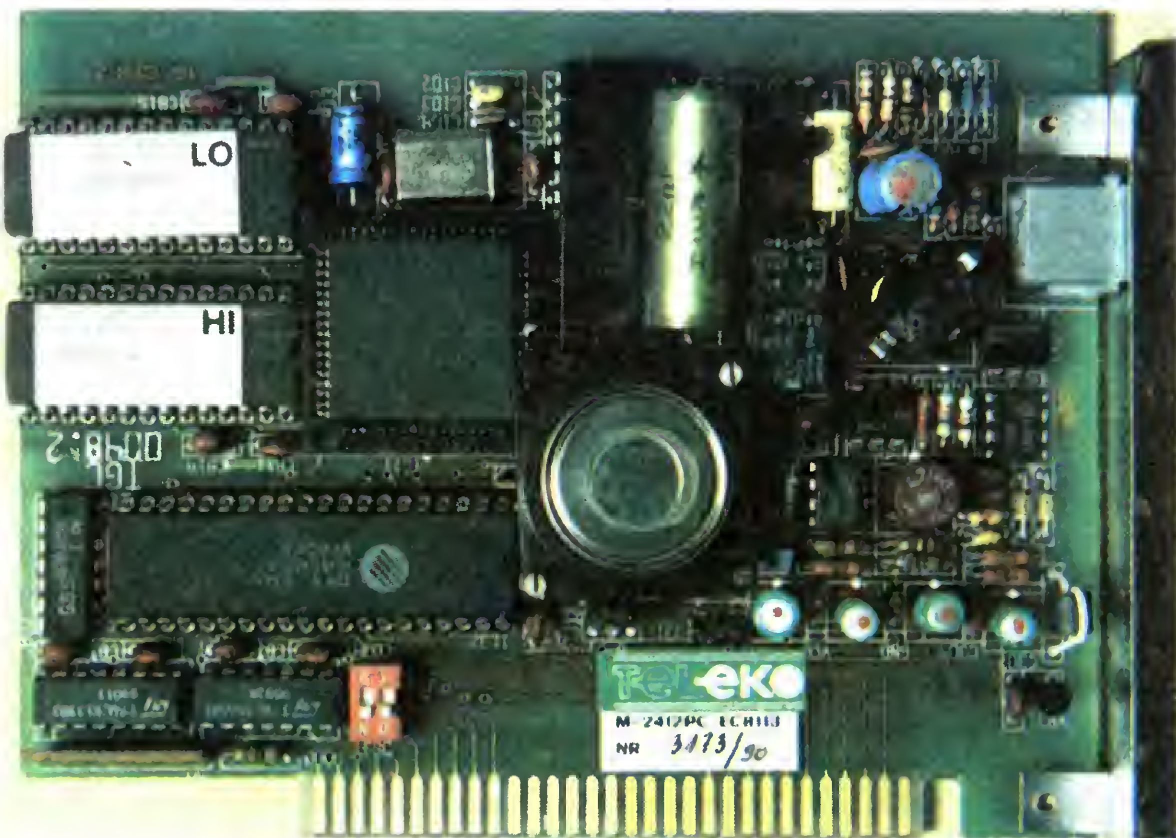
INSTRUKCJA

Stare powiedzenie komputerowców brzmi: gdy wszystkie sposoby zawiodą, przeczytaj instrukcję obsługi. Przy pracy z modemem musiałem kilka razy się do niej odwołać. Pierwszym odruchem było zawsze przerażenie.

Instrukcja nazywa się „Dokumentacja techniczno-ruchowa" ma 37 stron i zawiera opis modemu, dane techniczne i komplet podstawowych informacji dla modemiarza. Jednak konstrukcja i formuła instrukcji nie jest najszczęśliwsza. Informacje są zagmatwane, użyty język ściśle specjalistyczny, zaś co kilka stron straszą tabele bitów, rejestrów itp. Jest to na pewno potrzebne profesjonalistom, lecz prosty użytkownik gubi się w tym.

Ciekawość moją wzbudził również system oznaczeń modemów. Na okładce i we wstępie widać, że instrukcja dotyczy modemu EC 8113. Wczytując się można odszyfrować, że oznaczenie to dotyczy grupy dwóch modemów — M-2412 i M-2412 PC. Po co tyle symboli, cyfr i oznaczeń — nie wiem. Zgaduję, że oznaczenie 2412 to skrócony zapis dostępnych prędkości transmisji — 2400 i 1200 bodów.





NA KONIEC

Jak już podkreśliłem, fakt wyprodukowania w Polsce tak dobrego modemu uważam za znaczący. Jedyna odstraszająca cecha to cena. Jeśli nie może być tanio i dobrze, to lepiej dobrze niż tanio.

Czytelnikom chcącym zapoznać się z techniczną stroną modemów Tel-Eko, polecam test modelu M-2412 w „Bajtku" 6/91

Marcin Przasnyski

NAUKA?

Już od czasów średniowiecza utarł się pogląd, że człowiek uczony, to ten, który zna wiele faktów.

Przez długie lata takie twierdzenie miało dobre uzasadnienie, gdyż jeden człowiek mógł poznać i zapamiętać niemal cały wcześniejszy dorobek ludzkości. Jeszcze całkiem niedawno specjalista np. w dziedzinie fizyki znał wszystkie fakty, teorie i wzory ze swojej gałęzi wiedzy. Dziś nie jest już tak łatwo.

Dzisiaj niemożliwe jest poznanie całości dorobku ludzkości choćby w jednej dziedzinie. Powstają coraz węższe specjalności, a naukowcy coraz bardziej separują się nie tylko od reszty ludzi, ale nawet od siebie nawzajem.

Sytuacja taka wymuszona jest przez ogromne ilości faktów, doświadczeń i zagadnień związanych z ich pracą. Po prostu nie jest możliwe, by jeden człowiek poznał dogłębnie więcej, niż jedną dziedzinę.

Taki rozwój sytuacji musiał zmienić pojęcie człowieka uczonego. Przez pewien czas uważano, że najlepiej wiedzieć trochę o wszystkim. Jednak to „trochę" tak się rozrastało, że i ta droga skończyła się z przyczyn technicznych — jak często bywa w dzisiejszych czasach najsłabszym elementem układu okazał się człowiek. Długo i całkiem nieźle trzymała się wersja wąskiej specjalizacji. Powstały zastępy inżynierów, którzy uważali, że szklane domy są złą ideą ze względów technologicznych czy humanistów uważających Newtona za sprzedawcę jabłek.

Takie podejście do zagadnienia można przyjąć, gdy mamy do czynienia z ludźmi dorosłymi. W końcu ktoś musi tworzyć wiedzę, być najlepszym w swojej dziedzinie, być specjalistą. Ale wszystko to są ludzie, którzy świadomie *wybrali* wiedzę jako styl życia. Decydując się na swoją specjalność zrezygnowali z możliwości poznania innych dziedzin życia i godzą się na takie ograniczenia.

Poważnym problemem w takim podejściu jest odpowiedź na pytanie, czego uczyć dzieci. Wszystkiego po trochu? Programy szkolne są tak przeładowane, że nie tylko nie można już nic do nich dodać, ale należałoby skrócić je o połowę. Specjalizować dzieci już od najmłodszych lat? Znaczyłoby to pozbawić je możliwości wyboru (by bowiem wybrać trzeba wiedzieć co się wybiera).

Od pewnego czasu zdecydowano się na kompromis. W szkołach uczy się „przedmiotu" o nazwie „integrated sience". Te zespolone nauki to właśnie wszystkiego odrobinkę, za to dużo myślenia i wyciągania wniosków. Takie bowiem stało się nasze życie. Każdy problem (naturalnie poza kopaniem dołów) wymaga nie tyle wiadomości z różnych dziedzin wiedzy, ile wiadomości, *gdzie* można znaleźć rozwiązanie i jaka gałąź nauki zajmuje się naszym problemem. Wiadomości, fakty i dane można bez problemów uzyskać (jeśli wiemy gdzie ich szukać).

Niestety takie podejście do zagadnienia, jak wiele rzeczy u nas, występuje „za górami za lasami". U nas najlepszym rozwiązaniem problemów poznania współczesnego świata jest nadal dodawanie nowych faktów do przedmiotów szkolnych. Zupełnie, jakbyśmy chcieli wychować sobie bazy danych, a nie myślących ludzi. I nikomu nie przychodzi do głowy, że najlepszą bazą danych może być jeden z programów opisywanych na innych stronach pisma, a niekoniecznie musi być nią nasze dziecko. Odrobinę nadziei budzą możliwości wprowadzania przez nauczycieli programów autorskich. Daje to bowiem szansę, że o tym czego będą się uczyły dzieci zdecydują fachowcy, a nie urzędnicy. Powinny to robić uniwersytety, ale panuje powszechna opinia, że dydaktyka nie jest „prawdziwą" wiedzą i duża część naukowców traktuje katedry dydaktyki jak zesłanie, lub co najwyżej miejsce, w którym dostają pieniądze, a zajmują się inną (?), naukową nauką.

Miałem niedawno przyjemność oglądać, a następnie długie godziny spędzić przy grze komputerowej. Gra niestety była kradziona (złodzieje żegnali się z użytkownikiem napisem „do zobaczenia na giełdzie na Grzybowskiej")[1]. Gra nosiła tytuł SIMCITY i była grą strategiczną.

Grający otrzymuje na początek kwotę $20000 i dziewiczy, nie dotknięty nogą człowieka teren położony nad zatoką. Jego zadaniem jest zbudowanie tam miasta. Prawdziwego miasta dla jak największej ilości mieszkańców.

Jak każdy burmistrz, może grający wyznaczać i uzbrajać tereny pod zabudowę mieszkaniową i centra handlowe, przeznaczać działki dla zakładów przemysłowych, wytyczać drogi, koleje czy np. zbudować elektrownię (jądrowa kosztuje $5000 — to są widocznie bardzo cenne dolary).

Dlaczego jednak warto o tej grze pisać w klanie edukacji? Wprawdzie jej opis znajdzie się w piśmie TOP SECRET, ale będzie on dotyczył innych jej aspektów. Tam odsyłamy wszystkich, którzy chcą dowiedzieć się, jak walczyć z tornadem, co robić, by nie zdjęto nas z urzędu burmistrza, lub dlaczego jeszcze przed założeniem miasta musimy postawić elektrownię i bardzo dokładnie zaplanować kierunki rozwoju.

My zajmiemy się innymi aspektami tej gry: jej możliwościami kształcącymi. Czy grając w nią możemy się czegoś nauczyć? W rozumieniu naszego szkolnictwa — nie. Nie poznamy ani zasad dynamiki Newtona, ani żadnego wzoru chemicznego, ani nawet daty bitwy pod Grunwaldem. Jedynym ukłonem w stronę tak pojmowanej wiedzy jest możliwość poznania nazw kilku miast (np. Tokio), którymi możemy zarządzać, jeśli nie chcemy budować miasta od podstaw. Z tego punktu widzenia należałoby grę usunąć o kilometr od najbliższej ze szkół i nie dopuszczać do niej uczniów.

Czego zatem można przy jej pomocy nauczyć? Na pewno znajdziemy tu elementy wielu dziedzin życia i wiedzy, jednak nie są one w sposób jawny powiązane z konkretnymi przedmiotami szkolnymi.

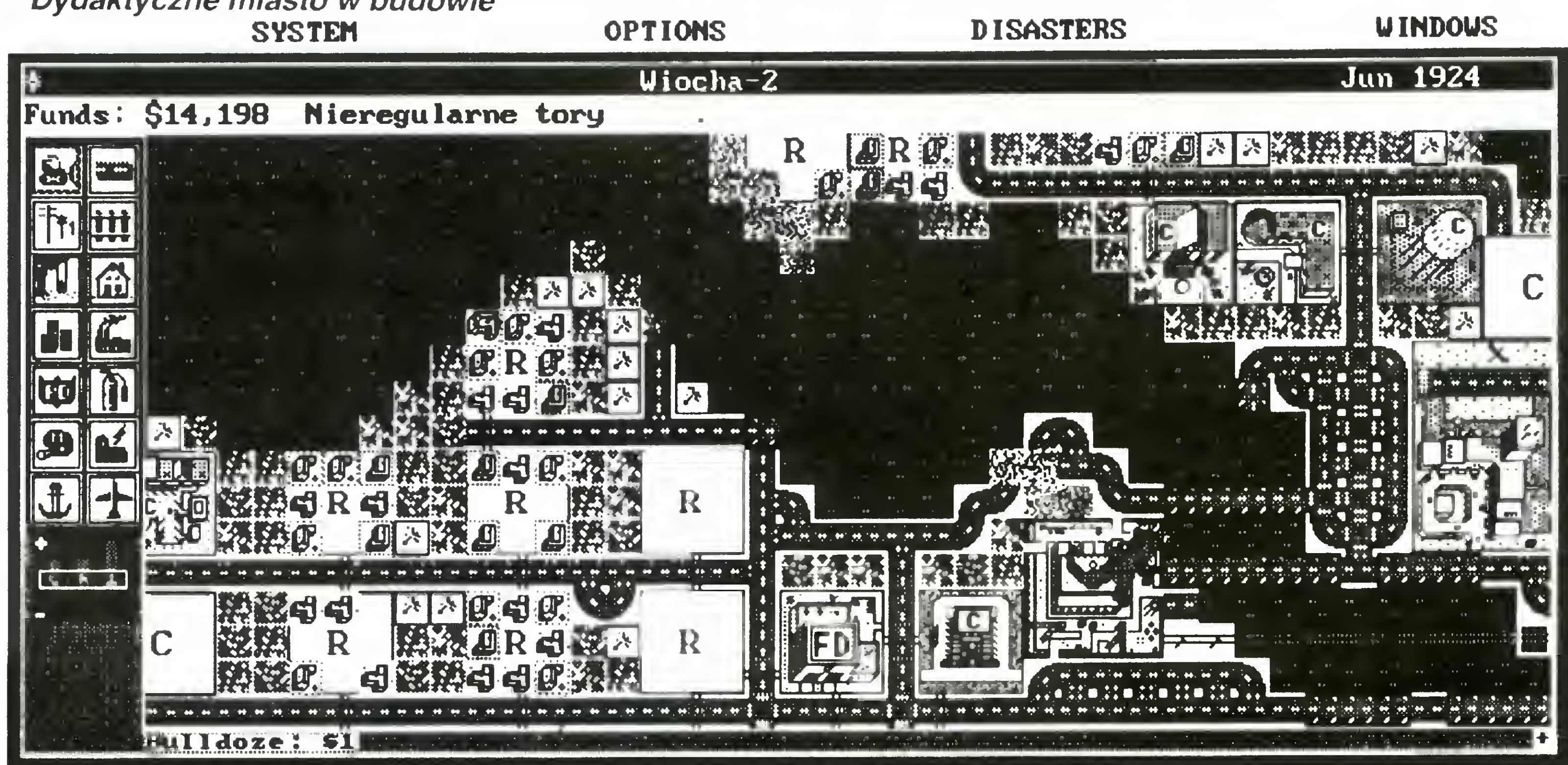
W toku gry wychodzą różne zależności geograficzno-socjologiczne i z pewnością pierwsze budowane miasto różnić się będzie od następnych. A to oznacza przecież zdobycie przez grającego wiedzy w zakresie planowania przestrzennego.

Miasto zaczynamy budować od elektrowni. Bez prądu przecież ani nie powstanie przemysł, ani nikt się nie sprowadzi. Następnie wytyczamy piewsze tereny pod budownictwo jednorodzinne i centra handlowe. Po uzbrojeniu terenu (prąd!) okazuje się, że ... najpierw powstają sklepy, a dopiero potem domy. Ale co się dziwić: program powstał na zachodzie i autorzy odwzorowują tamtejsze realia (nigdy przecież nie widzieli Ursynowa). Po zagospodarowaniu pierwszych obszarów należy szybko tworzyć zakłady pracy. Tu dają o sobie znać pierwsze trudności. Albo wybudujemy je za blisko dzielnic mieszkaniowych i wówczas mieszkańcy będą emigrować z powodu zanieczyszczenia, albo za daleko i będziemy musieli bardzo rozbudować sieć dróg dojazdowych (których utrzymanie słono kosztuje). Drogi muszą być jednak utrzymywane w dobrym stanie, bo inaczej ludzie będą się wyprowadzać, a biznesmeni niechętnie inwestować na naszym terenie. To jest w zasadzie pierwsza lekcja jaką otrzymuje gracz: władze muszą dbać o ludzi i organizować życie tak, by wygodnie było mieszkańcom a nie władzy.

Warto zwracać uwagę na zieleń miejską. Otoczenie dzielnic mieszkaniowych parkami bardzo przyspiesza zasiedlenie, co z kolei zasila budżet wpływami z podatków. Szybko również zasiedlane są tereny nad wodą, i takie dzielnice, gdzie jest dużo centrów handlowych. Niby oczywiste, ale jakże chętnie wprowadziłbym tę grę jako obowiązkową w polskich wydziałach gospodarki przestrzennej.

Gdy zakładane przez nas miasto rozrośnie się, należy zadbać o bezpieczeństwo mieszkańców. Posterunki policji

Dydaktyczne miasto w budowie





i straży pożarnej drogo kosztują, a utrzymanie tych instytucji jest bardzo drogie. Jednak nie można z tego zrezygnować. Duża przestępczość spowoduje niechybnie większą migrację ludności, a pożary prawie natychmiast unieruchomią przemysł. A dotacji z budżetu centralnego, bez których u nas nie można sobie poradzić, nie ma.

Bardzo ważna jest sieć dróg i kolei. Dobrze zaprojektowana przyspiesza zasiedlenie i rozwój przemysłu. Celowo nie wspominam o rozwoju handlu: jeśli wokół centrum hanldowego są zamieszkane dzielnice, to nie ma możliwości, by domy towarowe były zamknięte. A to znowu oznacza dochody dla budżetu.

Gdy nasze miasto zaczyna być duże, początkujący trafiają na kolejną trudność: prawie wszyscy stawiają elektrownie zbyt blisko planowanych dzielnic mieszkalnych. Prawie nikt nie podejrzewa, jak szybko i jak bardzo może rozrosnąć się miasto. A, żeby zasiedlić tereny wokół elektrowni trzeba się nieźle natrudzić (czytaj: dobrze dbać o mieszkańców).

Bardzo pobudzająco na rozwój miasta działa stworzenie portu czy lotniska. To rozwija przemysł, ludzie chcą mieszkać, gdy mają miejsca pracy, rozwija się handel. A budżet wciąż rośnie. Dzięki temu można zakładać nowe dzielnice i tak dalej.

Gdy nasze miasto staje się metropolią, przeważnie dostajemy trzecią lekcję myślenia: cały układ miasta przekroczył nasze oczekiwania i trzeba niemal całkowicie zmienić układu komunikacji. Doświadczeni burmistrze od początku planują z rozmachem, a początkujący muszą dopiero dorosnąć do myślenia wielkimi kategoriami. Przebudowa trwa długo, jest bardzo kosztowna, a jej uciążliwości powodują, że wielu mieszkańców przenosi się do sąsiednich miast. To z kolei powoduje... Ale tą lekcję już przerabialiśmy.

Po pewnym czasie mieszkańcy stają się wymagający: nie wystarcza już im, że mają ładnie zaprojektowane miasto. Życzą sobie np. stadionu. To kosztuje. Ale w końcu trzeba spełniać takie zachcianki, by mieć kim rządzić. Trzeba jednak mądrze planować: nie można wydawać wszystkich pieniędzy, lecz musi być zachowana pewna rezerwa na wypadek zdarzających się od czasu do czasu klęsk żywiołowych. Szkody wyrządzone przez tornado, trzęsienie ziemi czy pożar trzeba naprawić, a to kosztuje. Pewnego razu spłonęła cała dzielnica, gdyż budżetu nie było stać na wybudowanie nowej komendy straży, a wydatki na straż w poprzednich latach były mocno okrojone ze względu na inne potrzeby. Była to kolejna lekcja z zakresu gospodarki finansami. Jak sobie radzić z pożarami — o tym już w TOP SECRET.

Oddzielną lekcję z zakresu finansów dostajemy w ciągu całej gry. Podatki. Tylko u nas myśli się, że im wyższe podatki, tym lepiej dla budżetu. Po otrzymaniu lekcji wiemy, i jest to dla nas oczywiste, że podatki powinny być niezbyt duże. Gdy tylko zwiększymy je ponad społeczne oczekiwania, znika przedsiębiorczość, zamykają się zakłady pracy... patrz lekcja 2. Z kolei zbytnie obniżenie podatków powoduje małe wpływy do kasy miejskiej, a wydatki (drogi, policja, straż) są stałe i można je tylko trochę ograniczyć, mając na względzie konsekwencje. Dopiero płynna gospodarka podatkami może zapewnić nam ciągłość wpływów na odpowiednim poziomie nie wyludniając przy tym miasta.

Można się zastanawiać, czy wymienione możliwości mogą czegoś nauczyć — życie jest jednak trochę bardziej skomplikowane. Jednak nikt chyba nie zaprzeczy, że dłuższa styczność z programem pozwoli uzmysłowić sobie trudności i zadania czekające w dorosłym życiu. Co ważniejsze nie są to zadania typu „Z miasta A do miasta B jedzie pociąg z szybkością v. Obliczyć...", które w najmniejszym stopniu nie interesują ani uczniów, ani nauczycieli. Jak wiadomo zaś uświadomienie sobie problemu i jego werbalizacja jest pierwszym, bardzo ważnym etapem nauki.

Program pozwala na wyobrażenie sobie pracy planisty. Być może kolejne wersje będą bardziej rozbudowane. Gdyby np. do wyboru najlepszego miejsca budowy elektrociepłowni węglowej trzeba było znać nie tylko ilości zużywanej wody, węgla czy sposoby zagospodarowania odpadów, ale także potrafić zaplanować jej moc w zależności od sprawności sieci cieplnej, przewidzieć opory rurociągów w zależności od przekroju rur, czy rodzaj przewodnika stosowany w liniach wysokiego napięcia, to wartości dydaktycznych programu nie zakwestionowałby ani nauczyciel fizyki, ani biolog, ani geograf. A przecież taki program może powstać już dziś.

Nie warto więc zniechęcać dzieci do zabawy z komputerem. Niech, przynajmniej w czasie wakacji pograją sobie, a my cieszmy się, jeśli się przy tym czegoś nauczą. Zaś nocą, gdy dzieci już śpią, sami włóżymy dyskietkę i wpiszmy „SIMCITY". Pomoże nam to przynajmniej nabrać wprawy przy podejmowaniu decyzji, za której jesteśmy potem odpowiedzialni i których efekty obciążają nas samych.

T.B.Mańk

[1)] ***Pytanie konkursowe: Co to za kraj, w którym złodziej może podać nie tylko miejsce, gdzie sprzedaje swoje łupy i oficjalnie reklamować się we wszystkich, prócz BAJTKA pismach i nic mu za to nie grozi? MIĘDZY AUTORÓW PRAWIDŁOWYCH ODPOWIEDZI NADESŁANYCH DO KOŃCA WRZEŚNIA ROZLOSUJEMY JOYSTICK.***



IBM NA FIZYCE

Ostatnio na świecie furorę robi wykorzystanie komputera na lekcjach fizyki we wspomaganiu doświadczeń. Pisaliśmy już o tym w Bajtku 4/91, a teraz, tuż przed nowym rokiem szkolnym powracamy do tematu.

Jedni z pierwszych w Europie zajęli się tym Holendrzy Amsterdamska fundacja o nazwie CMA wypuściła na rynek (na razie holenderski, ale chcą również na polski) pakiet programowy o nazwie IP-COACH 3.0, który wraz z kartą do IBM, interfejsem i zestawem czujników stanowi kompletne wyposażenie wspomagania doświadczeń w pracowni fizycznej.

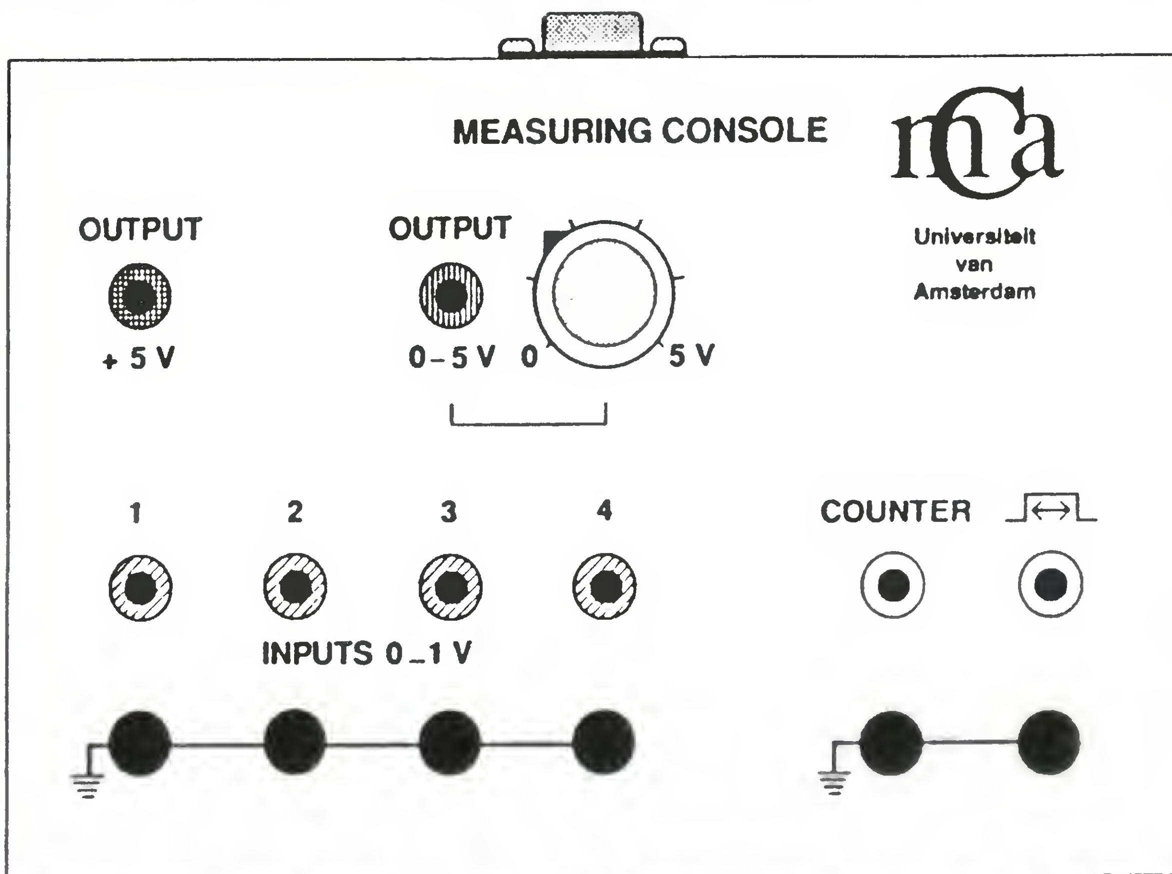
Przedstawiony na rys. 1. interfejs pomiarowy zawiera cztery niezależne wejścia analogowe, jedno wejście cyfrowe (wykorzystywane jako licznik), a prócz tego wyjście napięcia zasilającego +5V, jedno regulowane wyjście zasilające 0-5V, oraz generator impulsów. Wydaje się, że to bardzo dużo, choć w praktyce okazuje się, że trochę brakuje większej ilości wejść cyfrowych (które mogą być wykorzystywane dla bramek świetlnych).

Oddzielnie kupowane są sondy pomiarowe. W zasadzie wszyscy kupują miernik temperatury, światła, PH i ultradźwiękowy miernik odległości.

Po bezproblemowym podłączeniu do komputera i zainstalowaniu programu możemy przystąpić do prowadzenia doświadczeń. Jednak najpierw warto w całości przeczytać dość obszerną instrukcję obsługi. Nie tylko po to, by dowiedzieć się, jak obsługiwać program i planować doświadczenia. W przeciwnym razie z wielu opcji nie skorzystalibyśmy w ogóle, nie wpadlibyśmy bowiem na to, że można je zmieścić w niezbyt długim programie.

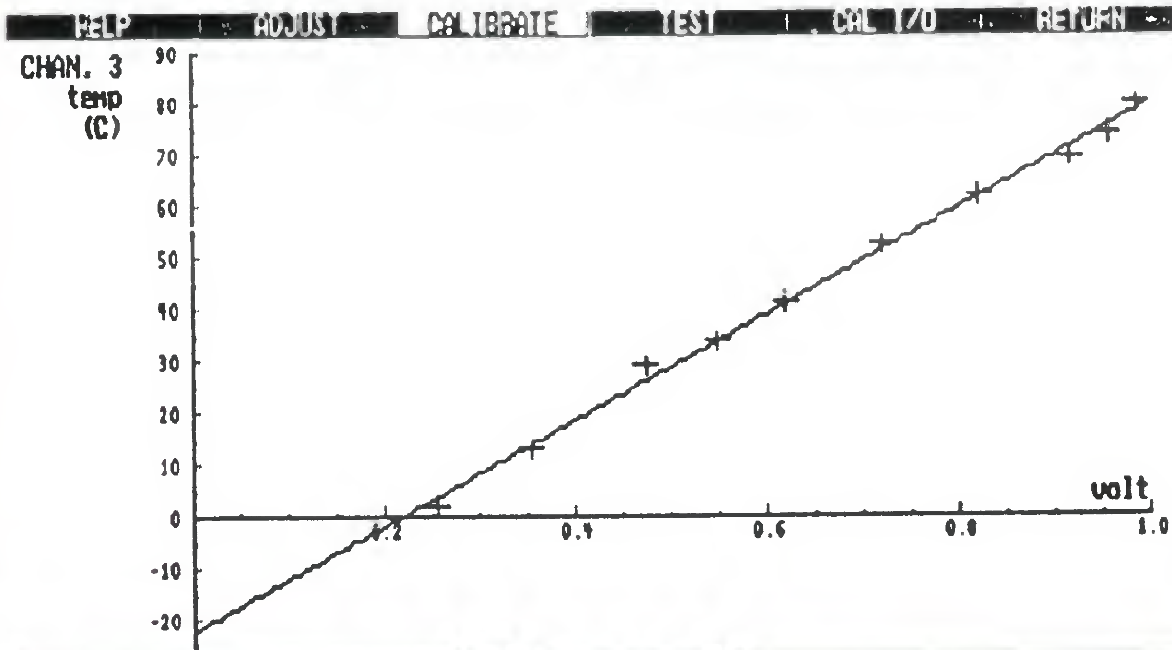
A jest tego sporo. Po pierwsze sondy trzeba wyskalować. Jeśli sonda jest liniowa, to wystarczy podać dwa punkty kontrolne, jeśli nie, wówczas program umożliwia nam zdjęcie całej charakterystyki.

Panel interfejsu.



Rys. 1)

Kalibracja sondy



Rys. 2)

Prezentacja wartości mierzonej



Rys. 3)

Oczywiście możemy to zrobić dwojako: bądź wprost mierząc wartości, bądź podając z klawiatury dane odczytane z metryczki sondy, czy innych pomiarów.

Po drugie możemy dokonać pomiarów. Dane mogą być zapamiętywane i przeznaczone do dalszej obróbki, bądź wielkość aktualnie mierzona może być wyświetlana (dużymi, wyraźnymi cyframi, widzianymi nawet z końca klasy) na ekranie.

Na razie są to oczywistości. Cóż wart byłby program pomiarowy bez możliwości skalowania sond i możliwości dokonywania pomiarów. Dopiero jednak analiza wyników doświadczenia pozwala w pełni wykazać moc programu. Dane można przedstawić w formie wykresu w najróżniejszych formach. Można posługiwać się wykresem funkcji pierwotnej i pochodnej, przedstwiać zależności wielkości mierzonych jednocześnie, czy liczyć pole pod krzywą. Jednocześnie na ekranie można zestawiać dwa różne wykresy przedstawiające różne zależności, co pozwala np. na porównanie krzywej i jej pochodnej, dwu wielkości mierzonych w tym samym czasie itp.

Inną, odrębną częścią analizy danych jest dopasowywanie krzywych do wyników doświadczeń. Do wyboru mamy kilka podstawowych typów krzywych. Po ustaleniu rodzaju zależności wystarczy, nanosząc kolejne punkty (deformując krzywą) dopasować ją do wykresu.

Bardzo pomocnym narzędziem wykorzystanym w programie jest blok poświęcony modelowaniu zjawisk fizycznych. Idea nie jest nowa, powstała kilkanaście lat temu w Anglii, a i w Polsce programy modelujące znane są od wielu lat. Jednak takie połączenie modelowania zjawisk i rzeczywistych doświadczeń stwarza nowe, olbrzymie możliwości nauczania fizyki.

Na czym polega modelowanie? Otóż jest to przedstawianie zjawisk fizycznych w postaci równań. Jednak nie są to równania znane ze szkoły. Są to raczej formuły pozwalające wyliczać kolejne wielkości w zależności od innych. I tak np. model ruchu jednostajnie przyspieszonego przedstawia się następująco:

F = const1 m = const2 dt = const3
a = F/m (przyspieszenie)
v = v + a∗dt (prędkość)
x = x + v∗dt (położenie)

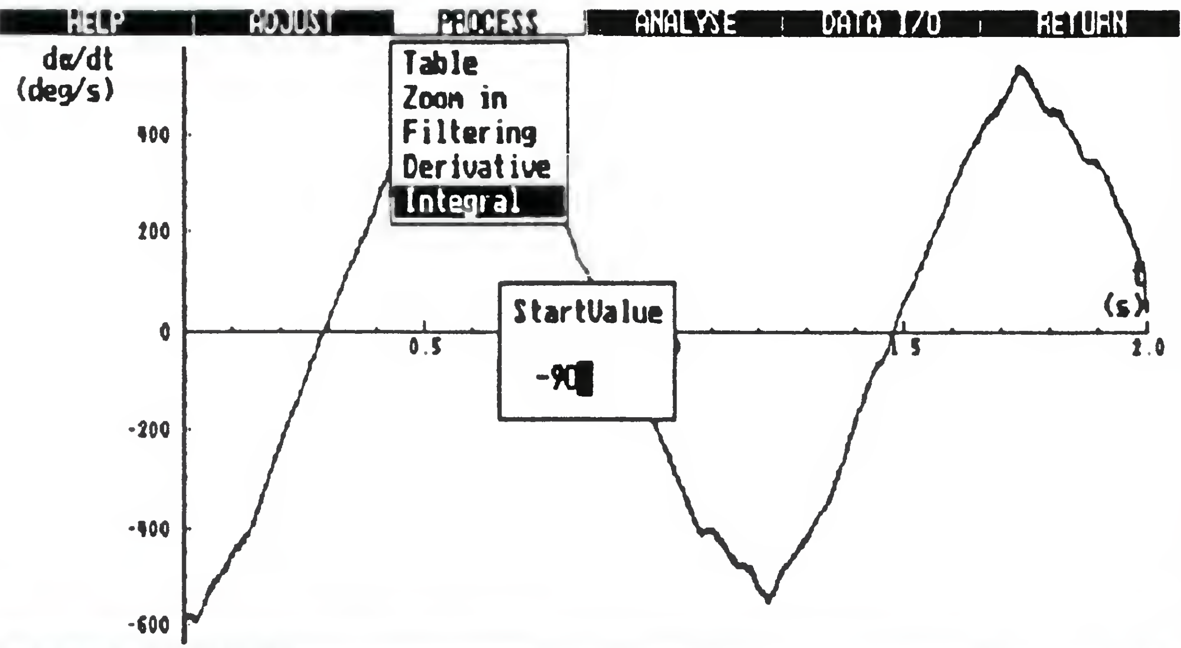
Wymienione równania wykonywane są iteracyjnie: po wyliczeniu nowego położenia obliczane jest nowe przyspieszenie (w ruchu jednostajnie przyspieszonym jest stałe), nowa prędkość i znowu nowe położenie itd. Z równań tych łatwo jest już przejść do zwykłych równań ruchu, a są one dużo bardziej przystępne dla uczniów.

W części programu odpowiedzialnej za modelowanie wystarczy wpisać równania i podać wartości początkowe i stałe. Program sam prześledzi historię układu i przedstawi nam w dowolnej postaci wyniki.

Jeśli teraz dokonamy porównania modelu z rzeczywistymi wynikami pomiarów, to każdy uczeń zrozumie, że używając takiego modelu (czyli konkretnych wzorów) można opisać dane zjawisko. Jest to tym bardziej oczywiste, że odbywa się niemal jednocześnie, na tym samym komputerze i w tej samej formie. Pokrywanie się modelu z rzeczywistością, przy dobraniu odpowiednich wartości początkowych jest wręcz uderzające i przemawia nawet do najmniej interesujących się fizyką. Omówienie całego programu, czy choćby wyliczenia wszystkich jego możliwości przekracza na pewno możliwości tego artykułu. Jednak niedługo będzie można obejrzeć ten program w kraju. Holendrzy bowiem są bardzo zainteresowani współpracą z Polską, nawiązali kontakty naukowe z kilkoma uniwersytetami i MEN-em, a nawet powstała spolszczona wersja programu.

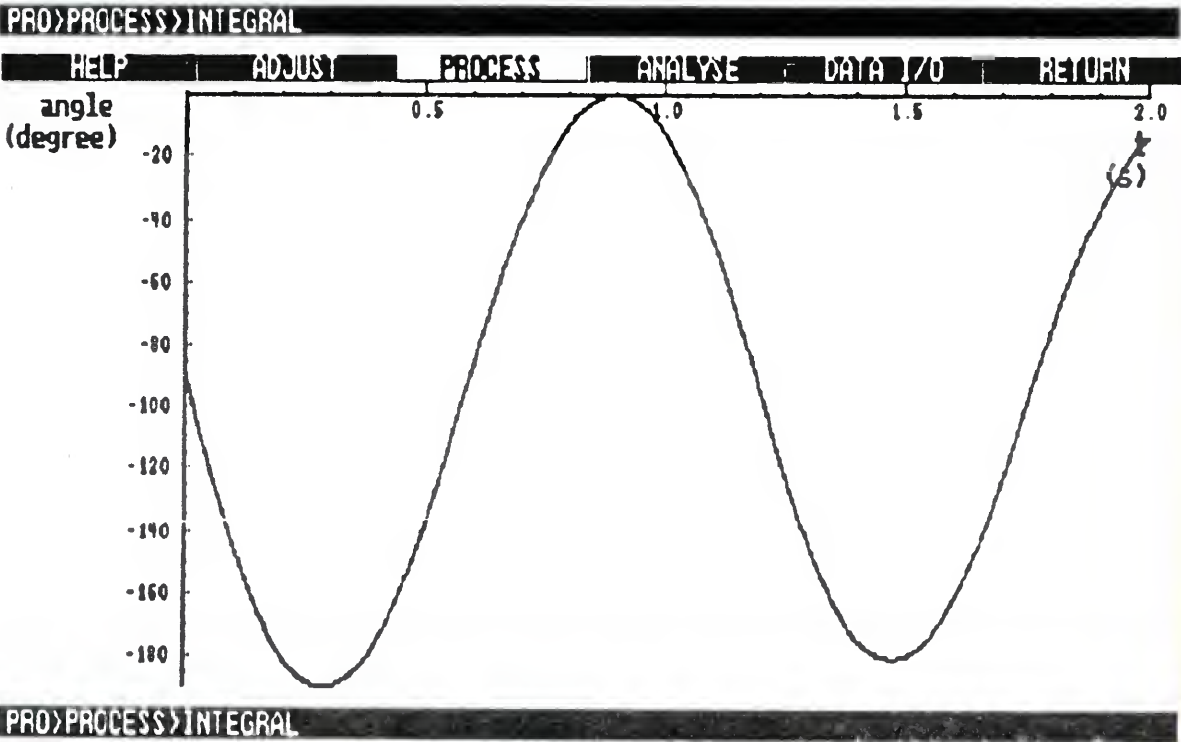
Trochę martwi tylko, że spora część naszych naukowców — dydaktyków traktuje zagranicznych kontrahentów jak statek przywożący dzikim paciorki. Owszem chętnie weźmiemy wasze towary, jak nam je dacie, bo jesteśmy biedni, a w zamian za to będziecie mogli nas nauczyć obsługi, najlepiej u was i na wasz koszt. Ale nie tylko ta dziedzina boryka się z pięćdziesięcioletnimi przyzwyczajeniami. W Bajtku 4/91 obiecaliśmy Państwu, że opiszemy polski miernik odległości powstający jako wynik współpracy Zakładu Edukacji Komputerowej z Bajtkiem i p. Grzegorzem Bujanowskim. Z przyjemnością informujemy, że sprawy poszły dalej niż planowaliśmy, i najpóźniej w październiku przedstawimy polski interfejs pomiarowy, którego miernik odległości będzie jedną z części składowych. Interfejs ten będzie taki sam, jak opisany przed chwilą, tylko lepszy. Może więc w Polsce można nie tylko dostawać paciorki.

T.B. Mańk



Dane pomiarowe

Rys. 4)



Dane pomiarowe z rys. 4 po operacji całkowania

Rys. 5)

CO INTERESUJE PRZEDSZKOLAKÓW?

W artykule tym chcemy Państwu przedstawić wyniki badań wstępnych nad zasadami odnoszenia się dzieci w wieku przedszkolnym do komputerów prowadzonych w Australii w roku 1987.

Wprawdzie Australia kojarzy nam się z czymś tak odległym, że przedstawianie wyników badań australijskich, jeśli nie dotyczą one kangurów, wydaje się nie na miejscu, jednak dzieci w wieku czterech lat są takie same na całym świecie.

1. METODA.

Typowe australijskie przedszkole zorganizowane jest w ten sposób, że na całym obszarze dostępnym dla dzieci zorganizowane są „kąciki zainteresowań" Dzieci mają swobodny wybór, czym chcą się zająć i jak spędzać czas. W trakcie badań po prostu dodano do innych kącik komputerowy. W nim dzieci były obserwowane, nie wiedząc o tym, co zapewnia wiarygodność wyników.

2. OBIEKT BADAŃ.

Badaniom poddane zostało 90-ro dzieci z przedszkola w Sydney. Średni wiek dzieci wynosił 4 lata i 4 miesiące, dzieci pochodziły ze średnio zamożnych rodzin mieszczańskich.

3. PROCEDURA.

Po okresie oswojenia dzieci z komputerem (i podstawowymi programami komercyjnymi (sic!)), dzieci będące w kąciku komputerowym były obserwowane podczas rozwiązywania dwu problemów: rysowania obrazków przy użyciu programu graficznego wzorowanego na języku pre-LOGO, oraz uproszczonej wersji LOGO, w której sterowanie żółwiem odbywa się przy użyciu pojedyńczych klawiszy. Oba zadania preferowały poszukiwania i były zgodne z programem przedszkolnym. Zachowanie dzieci przy komputerze było rejestrowane, a poszczególne przypadki zachowań były później klasyfikowane do jednej z 14 kategorii. Obserwacji podlegała również częstość i czas pobytu dzieci przy komputerze.

Kategorie, do których można było zaliczać epizody to:

A) pokazywanie/wyjaśnianie
B) omawianie/komentowanie
C) stawianie hipotez
D) śmiech
E) dzielenie się
F) cisza
G) fantazjowanie
H) pytanie
I) mówienie do siebie
J) prowadzenie wykładu
K) sprzeczka
L) odpowiadanie na pytania
M) uważne słuchanie
N) ogólna dyskusja

W trakcie trwających 6 tygodni badań zarejestrowano niemal 5000 epizodów, które dały się zakwalifikować do jednej z wymienionych grup.

4. WYNIKI.

W trakcie badań okazało się, że chłopcy dużo dłużej i częściej przebywali w kąciku komputerowym. Zauważono, że dzieci przychodziły do komputerów częściej w grupie, bądź parami niż pojedyńczo. Grupy najczęściej składały się z dzieci tej samej płci.

Procentowy rozkład zachowań przedstawiono na wykresie obok. Jak widać zdecydowanie przeważały dwa typy zachowań: omawianie/komentowanie oraz ... cisza. Łącznie do tych kategorii można było zaliczyć prawie połowę zachowań. Komentarze dzieci przeważnie dotyczyły tego, co aktualnie robią, dzieci nazywały kolory i figury, wzajemne ich położenie, liczyły. Duży odsetek ciszy w zachowaniach dzieci jest trochę mylący. Komputey nie zachęcają dzieci do tego, jak mogłoby wynikać z wykresu. Przy innym podejściu do zagadnienia okazuje się, że w innych sytuacjach, nie związanych z komputerem dzieci są ciche przez ok. 37% czasu, a komputer wymusza ciszę tylko na okres 30%. Widać więc, że wspólne spędzanie czasu z komputerem zwiększa raczej aktywność dzieci, niż ją zmniejsza.

Cieszyć może dość duży odsetek zachowań, które można zaliczyć do klasy stawiania hipotez. Dzieci przewidywały strategie, opracowywały plany *przed* przystąpieniem do działania. Świadczy to o wpływie komputera na sposób myślenia dzieci. Okazuje się też, że komputer może być doskonałą zabawą nawet dla tak małych dzieci, i nawet, gdy nie grają one w gry komputerowe. 9% zachowań, które można było zakwalifikować jako śmiech świadczy chyba o doskonałym humorze.

Z drugiego końca skali nie powinno nikogo dziwić, że słuchanie wykładów zajęło tylko 1% czasu. Wiąże się to po prostu z organizacją przedszkoli australijskich i powstrzymywaniem się wychowawców od ingerencji. Do wykładów dochodziło głównie na życzenie dzieci, a na początku dzieci zostały zapoznane z komputerem i jego możliwościami.

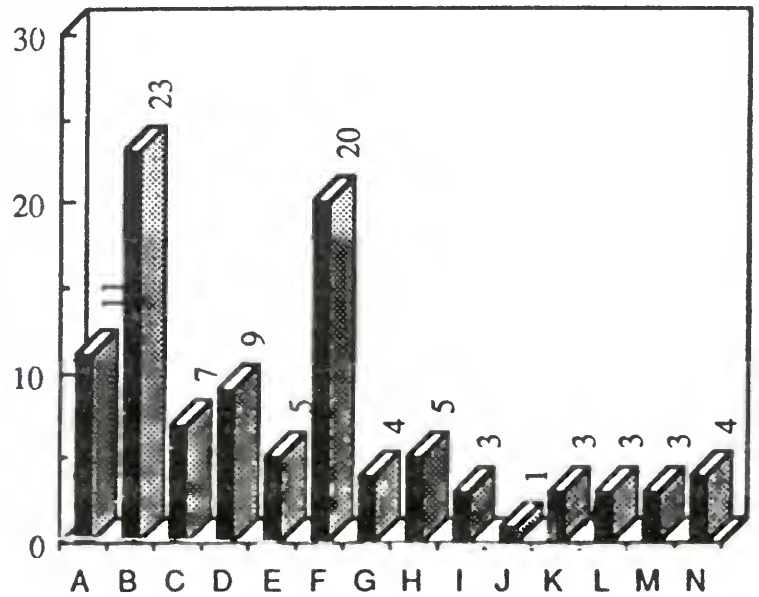
Sądząc z wyników badań, wprowadzenie komputerów do przedszkoli jest nie tylko możliwe, ale i niezbędne. Po pierwsze pozwoli to na naukę obsługi sprzętu, który będzie niezbędny w dalszym życiu. Po drugie zaś widać, że komputery wyrabiać mogą w tak małych dzieciach wiele zachowań pożądanych z punktu widzenia celów wychowawczych. Pozwalają one nauczyć dzieci stawiania hipotez, otwartego na problemy sposobu myślenia. Uczą również pożądanych społecznie zachowań, jak omawianie problemów, umiejętność wyrabiania sądu czy osiągania kompromisu. Warto również zwrócić uwagę, że aż 11% zachowań dzieci to było wyjaśnianie innym poznanych przez siebie umiejętności.

W Polsce są jeszcze ludzie, którzy niechętnie wypowiadają się na temat wprowadzania komputerów do szkół (chyba że wyższych). Prawdopodobnie więc nieprędko będzie można choć zacząć dyskusję nad komputeryzacją przedszkoli. Na świecie jednak taka dyskusja się toczy, a raczej toczyła. Teraz właśnie wskutek niej wprowadza się komputery w młodszych klasach i przedszkolach. I dzieci nie są przez to ani głupsze, ani mniej ludzkie, a za to są dużo lepiej przygotowane do życia.

T.B. Mańk

P.S *podane za: materiały z konferencji „COMPUTERS IN EDUCATION", Lousanne 1988. Badania przeprowadzono w The Uniwersity of Wollaongong*

Procentowy udział zachowań dzieci w różnych kategoriach



PRZENOSZENIE PLIKÓW MIĘDZY D-BASE II I D-BASE III

Wiele osób pracujących dawniej na komputerach ośmiobitowych, szczególnie na Amstradach CPC lub PCW zamieniło swój sprzęt na komputery szesnastobitowe — czy to Atari ST, Amiga, lub jakiś klon IBM PC. Często jednak pozostaje im ze starego komputera wiele dyskietek zawierających efekty wcześniejszej pracy — programy, pliki tekstowe z edytorów, bazy danych itp., które pragną „przepisać" na nowy komputer, aby dalej z nich korzystać.

Postaram się w niniejszym artykule przedstawić czytelnikom, jak dane i programy napisane w dBase II można przenieść i wykorzystać na komputerach IBM w programach dBase III lub dBase III Plus.

Informacje ogólne

Programy dBase II, dBase III i dBase III Plus są to kolejne wersje relacyjnych baz danych firmy Ashton-Tate. O ile nie ma (z punktu widzenia przenoszenia danych) specjalnych różnic między wersjami III i III Plus, o tyle wersja II różni się od nich dość znacznie. Ogólnie rzecz biorąc problemy dotyczą:

a) przeniesienia danych z używanego wcześniej komputera na IBM,
b) binarnej formy zapisu rekordów danych — różnej dla dBase II i dBase III,
c) innej postaci komend (zbiory komend i formatów .CMD i .FRM).

Użytkownicy oryginalnych, legalnych wersji programu dBase III lub III Plus mogą znaleźć wśród plików na dyskietce opatrzonej etykietą **Sample Files and Utilities** jeden, nazwany **DCONVERT.COM.** Jest to program dokonujący w pełni automatycznie wszystkich procesów dostosowania plików danych, komend i formatów do postaci akceptowalnej przez dBase III. Użytkownicy ci także posiadają podręcznik dołączony do programu i tam znajdą precyzyjny opis, jak program DCONVERT uruchomić i obsługiwać. Pozostaje wówczas jedynie problem przeniesienia pliku danych na użytkowany komputer.
Wszyscy pozostali muszą niestety dokonać zmian samodzielnie — poniżej opisane są konieczne do tego operacje.

Przenoszenie danych między Amstradem a IBM-em

W przypadku Amstrada CPC 6128 problem przenoszenia danych jest od dawna rozwiązany — istnieją programy pozwalające przenieść pliki z dyskietek Amstrada na dyskietki w formacie IBM. Jeden z takich programów — **TDOS 128** — został już opisany na łamach Bajtka, w jednym z ostatnich numerów, tam więc odsyłam wszystkich zainteresowanych. Osoby nie posiadające już „starego" komputera — mogą wykorzystać, już na nowym, specjalny program kopiujący, taki jak np. CONVERT. Pozwala on na wczytywanie plików z dyskietek różnych typów komputerów, między innymi także Amstrada CPC i zapisywanie ich na dyskach IBM.

Przenoszenie plików danych dBase'a

Tutaj jest już nieco większy problem. Pliki **.DBF** z dBase II nie są akceptowane przez dBase III. Spowodowane jest to inną strukturą binarną nagłówka, wynikającą z różnic możliwości kolejnych wersji (ilość rekordów, długość pól itp.). Problem da się jednak rozwiązać zupełnie łatwo, poprzez mechanizm obsługi przez dBase plików tekstowych.

Plik danych (.DBF) zawiera w nagłówku informację o strukturze utworzonej przez użytkownika bazy danych (nazwa, typ i długość pól opisujących rekord danych itp.) Podczas tworzenia plików danych w postaci tekstowej te właśnie informacje ulegają skasowaniu — tekstowa wersja danych **nie zawiera informacji o strukturze całej bazy!** Konieczne jest więc odczytanie struktury bazy danych, którą zamierzamy przenieść i utworzenie w dBase III bazy o **identycznej** strukturze rekordów. Strukturę rekordów w dBase II możemy odczytać komendą:

DISPLAY STRUCTURE

Po tej komendzie zostaną wyświetlone wszystkie informacje dotyczące budowy bazy danych — ilość rekordów, nazwy, długość, typ poszczególnych pól w rekordzie. Otwierając bazę danych w dBase III tworzona struktura rekordów musi być identyczna ze „starą" bazą. Do tak utworzonego „szkieletu" można już przepisać całą zawartość tekstowego pliku danych — w efekcie otrzymujemy pełny plik danych już w standardzie dBase III.

Postać tekstową pliku danych możemy otrzymać poprzez podanie komendy:

COPY TO [nazwa-pliku] SDF

Komenda ta przepisze zawartość bazy danych będącej w użyciu (USE) do nowo utworzonego pliku o nazwie [nazwa-pliku] z rozszerzeniem .TXT, na przykład

USE TEST
COPY TO TEMP SDF

Spowoduje utworzenie na dysku pliku TEMP.TXT i przepisanie do niego jako zbioru tekstowego wszystkich danych umieszczonych w pliku TEST.DBF.

Tak otrzymany plik możemy przenieść na dysk IBMowski i (po wcześniejszym stworzeniu bazy danych o strukturze identycznej z bazą przenoszoną) wczytać do dBase-a komendą:

APPEND FROM
[nazwa-zbioru].TXT SDF

Otrzymana w efekcie baza danych będzie zawierała wszystkie informacje zawarte w pierwotnym pliku dBase II.

UWAGA: Jeszcze raz podkreślam konieczność utworzenia bazy danych o identycznej strukturze — inaczej dane zostaną wczytane niepoprawnie!

Modyfikacje plików komend i formatów

W związku ze zdecydowanym rozbudowaniem możliwości bazy danych dBase III w stosunku do wersji II postać wielu komend została zmieniona lub rozszerzona. Konieczna jest w związku z tym modyfikacja plików **.CMD.** Poniżej przedstawię większość dokonanych zmian postaci komend.

1) Zmiana separatora wyrazów.

Zamiast znaku dwukropka : rozdzielającego wyrazy w opisie pola lub zmiennej stosowany jest znak podkreślenia _ . Jeśli więc zmienna nazwana była np:

DATA:URODZ

to teraz musi wyglądać:

DATA_URODZ

2) Zmiana postaci komend SELECT PRIMARY i SELECT SECONDARY.

dBase III pozwala na otworzenie do 10 baz jednocześnie (gdy tymczasem dBase II pozwalał na pracę najwyżej w dwu). Mechanizm wyboru bazy musiał więc być dostosowany do nowych potrzeb — selekcja następuje poprzez podanie nie nazwy (PRIMARY,SECONDARY) lecz numeru, np:

SELECT 1
SELECT 2
...
SELECT 10

Tak więc komendy SELECT PRIMARY i SELECT SECONDARY należy zastąpić odpowiednio komendami SELECT 1 i SELECT 2.

3) *Zmiana wskaźników do plików.*

W dBase II można wybierać jedynie pomiędzy plikami PRIMARY i SECONDARY. Jeśli nazwa pola występowała w obydwu bazach — wskazywana była prefiksem **P.** *lub* **S.** , *np:*

P.IMIE
S.ADRES

Ze względu na możliwe otworzenie większej ilości baz został stworzony inny mechanizm dostępu. Prefiksem może być nazwa pliku, litera A do J lub zastępnik nazwy (ALIAS). Za prefiksem zapisywany jest wskaźnik → (znak przeniesienia a za nim znak większości) i opis pola rekordu, np:

PRACE → PLACA

gdzie PRACE — to nazwa pliku, PLACA — nazwa pola w rekordzie danych tego pliku.

Przydzielanie zastępników nazwy następuje poprzez rozszerzoną postać komendy USE — patrz poniższy przykład:

SELECT 1
USE IMIONA ALIAS F1
SELEKT 2
USE PRACE ALIAS F2
SELECT F1
? F2 → PLACA

Konieczna jest więc w plikach dBase II zmiana prefiksu S. lub P. na odpowiedni wskaźnik. Najwygodniej — aby zapewnić poprawne działanie programu — wykorzystać prefiks literą (A,B odpowiednia dla plików PRIMARY i SECONDARY).

dBase II	dBase III
? S.IMIE	?B → IMIE

UWAGA: Z możliwości opisu bazy jedną litera (dBase III) wynika ograniczenie — **nie wolno tworzyć plików o nazwie opisanej jedną literą w zakresie A–J. Są one zastrzeżone dla komendy ALIAS!**

4) Zmiana postaci wskaźnika numeru rekordu.

Wskaźnik numeru rekordu # został zastąpiony poprzez funkcję **RECNO ().**

5) Zmiana postaci wskaźnika skasowania rekordu.

Wskaźnik skasowania rekordu * został zastąpiony poprzez funkcję **DELETED ().**

6) *Zmiana postaci funkcji wskazującej koniec zbioru.*

Wskaźniki końca zbioru **EOF** został zastąpiony poprzez funkcję **EOF ().**

7) Zmiana nazwy funkcji operujących na łańcuchach.

dBase II	dBaseIII
@	AT
!	UPPER
$	SUBSTR
RANK	ASC

Postać parametrów nie ulega zmianie.

8) Zmiana klauzuli USING.

W komendzie SAY klauzula **USING** została zastąpiona klauzulą **PICTURE**.

9) Zmienne logiczne muszą być inicjalizowane wyłącznie do postaci .T. lub .F. (w dBase II dopuszczalne jest inicjalizowanie do **T** lub **F**, tj. bez kropek).

10) Ze względu na traktowanie zmiennych jako lokalnych (w dBase III) konieczne jest umieszczenie klauzuli PUBLIC w programie głównym.

Ponadto niektóre komendy dBase II nie mają swoich odpowiedników w dBase III. Poniżej w tabelce przedstawione są zmiany postaci komend w wersjach dBase II i III. Jeśli przenoszony program wykorzystuje którąś z tych funkcji koniecznym jest zastąpienie jej najbliższymi odpowiednimi komendami dBase III.

KOMENDY I ICH ODPOWIEDNIKI

dBase II	dBase III
# (numer rekordu)	RECNO ()

* (test skasowania)	DELETED ()
@ ()	AT ()
$ ()	SUBSTR ()
! ()	UPPER ()
RANK ()	ASC ()
CLEAR	CLEAR ALL
DISPLAY FILES	DIR
ERASE	CLEAR
QUIT TO	skasowana, zobacz RUN/!
READ NOUPDATE	skasowana, zobacz READ
REMARK	skasowana, zobacz ? i @ .. SAY
RESET	skasowana, zbędna w DOSie
SET COLON ON	skasowana, zobacz SET DELIMITERS
SET DATE TO	skasowana, zobacz DATE (DOS)
SET EJECT ON/OFF	skasowana, zobacz REPORT
SET HEADING TO	skasowana, zobacz REPORT
SET RAW ON/OFF	skasowana, zobacz TRIM ()
SET SCREEN ON/OFF	skasowana, zobacz SET DEVICE i SET CONSOLE
TEST	skasowana, zobacz TYPE ()
UPDATE .. ADD	skasowana, zobacz UPDATE

Konwersja plików formatów

Ze względu na inną postać zapisu (tekstowa w dBase II, binarna w dBase III) pliki **FRM** utworzone pod dBase II nie mogą być bezpośrednio wykorzystywane w dBase III! Jedynie program DCONVERT umożliwia przetworzenie pliku formatów do postaci wykorzystywanej w dBase III. Jeśli go nie posiadamy — pliki te muszą być ręcznie utworzone w dBase III.

Jeszcze kilka uwag o zmianach...

Jest jeszcze jeden problem związany ze zmianami działania poniektórych komend. Niektóre ze wskaźników są modyfikowane podczas pracy programu nieco inaczej. Na przykład komenda **FIND** w dBase II jeśli nie odnajdzie rekordu — ustawia wskaźnik numeru rekordu na **0**. W dBase III wskaźnik ustawiany jest na koniec pliku (ostatni rekord), aby zapewnić zgodność działania komendy FIND z komendą LOCATE, oraz ustawiany jest wskaźnik **EOF ()** na wartość logiczną TRUE. Podobnie, jeśli wykonujemy komendę **SKIP –1** gdy wskaźnik rekordu jest ustawiony na rekordzie pierwszym — w dBase II jest ustawiany na **0**. W dBase III natomiast wskaźnik w dalszym ciągu jest ustawiony na rekordzie pierwszym, natomiast nowy wskaźnik — **BOF ()** (Beggining Of File — początek zbioru) ma nadawaną wartość TRUE.

Podsumowanie

Jak więc widać, przeniesienie kompletnego programu z dBase II do dBase III nie jest tak bardzo skomplikowane. Nawet napisanie programu dokonującego automatycznie opisanych zmian dla dobrego programisty nie będzie nastręczało większych problemów. Zainteresowanym polecam doskonale opracowaną książkę **dBase Bridge — dBase III Plus** wydaną przez firmę Ashton Tate przedstawiającą szczegółowo różnice pomiędzy dBase II, dBase III i dBase III Plus i przechodzenie z jednej do drugiej. Na koniec przedstawię jeszcze tylko podstawowe dane techniczne baz danych dBase II i dBase III:

	dBase II	dBase III
ilość pól w rekordzie:	32	128
ilość znaków w rekordzie	1000	4000
ilość rekordów:	65535	nieograniczona
liczba otwartych baz:	2	10
precyzja obliczeń (cyfr)	10	16
maksymalna liczba zmiennych pamięciowych:	64	256

Stanisław Szczygieł

ARCHIWERY POD CP/M

Nie raz i nie dwa pojawiają się kłopoty z rzadko używanymi plikami. Zajmują one cenne kilobajty na dysku i nie zawsze można je wykasować.

Dobrym wyjściem jest archiwizacja plików, czyli ich skompresowanie*. Dzięki temu będą one zajmować mniej miejsca. Pomysł jest doskonały, jeśli dostępny jest odpowiedni program, zwany archiwerem lub pakerem. Do niedawna jedynym takim programem znanym w Polsce był NewSweep, posiadający opcję SQUEEZE. Kompresja tym algorytmem jest jednak mało efektywna.

Pod koniec zeszłego roku Staszkowi Szczygłowi udało się ściągnąć z Anglii kilka archiwerów. Są to programy ARK (trzy wersje) i ARC, UNZIP oraz UNARC. Różnice między programami ARK a ARC ograniczają się do nazw plików archiwalnych — ARK chce pliki z rozszerzeniem .ARK, zaś ARC — oczywiście woli .ARC. UNARC nie zwraca na takie detale uwagi i obsługuje zarówno pliki ARC jak i ARK. Osobną sprawą jest UNZIP — służy on do rozpakowywania archiwów stworzonych przez program PKZIP (na IBM PC i klonach, do wersji 1.10).

ARC i ARK

Jak dotąd, pracowałem z trzema wersjami programu ARK: ARK 1.1 (dla CP/M 2.2) oraz dwoma typami ARK 0.4 (jedna jest dla systemu TurboDOS, druga — CP/M Plus). ARC-a znam w tylko jednej wersji: ARC 2.4.

Wszystkie te wersje są ze sobą kompatybilne, a co więcej — są one kompatybilne ze znanym programem SEA ARC (pierwszy popularny archiwer na IBM PC), co oznacza jednocześnie kompatybilność z programami PKARC/PKPAK oraz PAK (po włączeniu odpowiedniej opcji — dla programu PAK jest to „/C").

Obsługa programu ARK jest bardzo prosta. Gdy widoczny jest systemowy „prompt" (znak zachęty), czyli „A>", trzeba wpisać

ARK archiwum plik1.typ plik2.typ ...

gdzie „archiwum" jest nazwą pliku archiwalnego, do którego chcemy dołączyć wymienione pliki („plik1.typ", „plik2.typ"). W nazwie pliku źródłowego może wystąpić znak „*" (tzw. wildcard), np „*.PAS" będzie oznaczać „wszystkie pliki z rozszerzeniem PAS". dodatkowo, nazwę archiwum można poprzedzić znakiem „-" (minus) i jedną z czterech opcji:

-B — zakazuje kasowania starej wersji archiwum

-K — zmusza do kompresji algorytmem CRUNCH niezależnie od tego, czy plik daje się skompresować

-S — zabrania kompresji, pliki są dołączane do archiwum w postaci nieskompresowanej

-W — wyłącza wszelkie komunikaty ostrzegawcze.

Aby włączyć dwie (lub więcej) opcje jednocześnie, np. B i K, należy napisać **„-BK"**. Jeśli wywołamy program bez parametrów, wyświetli on spis opcji oraz format parametrów.

Programy ARC i ARK używają trzech algorytmów kompresji (spośród ośmiu stosowanych przez SEA ARC).

— **STORE** — włączenie bez kompresji

— **PACK** (czasem nazywana RLE — Run Lenght Encoding) — polega ona na eliminacji ciągów jednakowych bajtów

— **CRUNCH** (algorytm LZW — Lempel-Ziv-Welch, od nazwisk autorów) — bardzo efektywny i dość skomplikowany.

Jeśli nie została użyta opcja **K** lub **S**, to program automatycznie wybiera najlepszy dla danego pliku algorytm.

Programy ARC/ARK mają jeden, mocno denerwujący, błąd: nie „zauważają" one braku miejsca na dysku i usiłują dalej zapisywać swoje dane. Efektem jest bezsensowny plik o rozmiarze kilku megabajtów (na dysku 180K) — nie trzeba chyba dodawać, że są to same śmiecie...

UNARC

UNARC jest programem do operacji na archiwach. Służy on do przeglądania zawartości archiwów oraz ich rozpakowywania. Oryginalna wersja programu UNARC 1.6 ma postać samorozpakowującego archiwum i zawiera dokumentację oraz dwie wersje programu: UNARCA dla systemów opartych na procesorze INTEL 8080 (i kompatybilnych, np 8085) oraz UNARC, dla systemów z procesorem Z-80 i kompatybilnymi (np. U880D).

Podstawowym zadaniem programy UNARC jest, jak sama nazwa wskazuje, rozpakowywanie. Może on rozpakowywać na trzy sposoby: na ekran, na drukarkę i do pliku. Dwa pierwsze sposoby mają sens tylko wtedy, jeśli rozpakowywujemy plik tekstowy — jeśli chcemy wydrukować plik binarny, UNARC protestuje.

Wywołanie programu UNARC może mieć postać

UNARC archiwum

takie wywołanie spowoduje wyświetlenie „spisu treści" archiwum, czyli listy plików (z podaniem rozmiaru przed i po kompresji, daty dołączenia do archiwum, sumy kontrolnej oraz procentowego wskaźnika stopnia kompresji). Jeśli archiwum zawiera więcej niż jeden plik, to dodatkowo wyświetlona zostanie informacja o ilości plików ich łącznej długości (przed i po kompresji) oraz wynikowym wskaźnikiem spakowania.

Wywołanie

UNARC archiwum plik opcja

spowoduje wykonanie na jednym z zawartych w archiwum plików operacji określonej przez opcję:

— bez opcji — wyświetlenie pliku na ekranie. Jednocześnie wyświetlane są 24 linie tekstu.

— opcja **N** — jak wyżej, ale bez zatrzymywania się.

— opcja **C** — sprawdzenie poprawności pliku (zgodności z sumą kontrolną).

— opcja **P** — jak opcja **N**, ale na drukarkę.

Jednorazowo można podać tylko jedną opcję i jedną nazwę pliku. Podczas drukowania lub sprawdzania zawsze podawane są (na początku) dane o pliku lub plikach. Nazwa pliku może zawierać znaki „*" i „?" — można w ten sposób wybrać kilka (lub wszystkie) pliki jednocześnie.

Trzecia forma wywołania ma postać

UNARC archiwum d:plik

Pozwala to na rozpakowanie pliku i nagranie go na dysk. Wymagane jest tu podanie nazwy dysku docelowego. Tak jak w poprzednich wywołaniach, otrzymujemy na bieżąco informacje o rozpakowywanych plikach.

KTO, KIEDY...

Program:	ARK 0.4P, ARK 1.1, ARC 2.4
Autor:	Brian E. Moore
Rok:	1989 (najnowsza wersja)
Program:	UNARC 1.6
Autor:	Robert A. Freed
Rok:	1987
Program:	UNZIP 1.20
Autor:	DPG
Rok:	1990

EFEKTYWNOŚĆ KOMPRESJI danych programem ARC v2.4

RODZAJ DANYCH	**STOPIEŃ KOMPRESJI** (max)
teksty	30% do 50%
teksty źródłowe	30% do 70%
binarne (programy)	5% do 30% *
grafika (obrazki itp.)	30% do 95%

Uwaga: Podane wskaźniki efektywności są przybliżone. W szczególnych przypadkach mogą wystąpić znaczne różnice. Podane dane pochodzą z dokumentacji programu.

* Zależy to od rodzaju programu. Dla konkretnych programów stopień kompresji może być większy.

TROCHĘ TERMINOLOGII

Do zrozumienia tego artykułu konieczna jest znajomość kilku terminów, które dotąd nie pojawiły się w „Klanie Amstrada".* Oto one:

KOMPRESJA — zmniejszanie rozmiaru danych/programu, zwykle przez specjalne kodowanie powtarzających się ciągów bajtów.

ARCHIWER — program dokonujący kompresji i tworzący archiwum. Niektóre archiwery potrafią także rozpakowywać archiwa.

ARCHIWUM — lub plik archiwum: plik zawierający skompresowane dane/programy, archiwum może zawierać wiele skompresowanych plików.

PAKER — patrz archiwer.

PAKOWANIE — kompresja dokonywana przez archiwer

ROZPAKOWANIE — dekompresja archiwum, odtworzenie nieskompresowanej postaci plików danych/programów zawartych w archiwum.

* Algorytmy kompresji opisano w Bajtku 7/91

```
A>arkp
ARK version 0.4
Brian E. Moore
CP/M Plus version
Usage: ARK [-bksw] <archive> [<filename> . . .]
Where:    b   = retain backup copy of archive
          k   = force compression (crunch)
          s   = suppress compression (store only)
          w   = suppress warning messages


A>unarc

UNARC  1.6  27 Mar 87
CP/M Archive File Extractor

Usage:  UNARC [d:]arcfile[.typ] [d:][afn] [N|P|C]

Examples:
B>UNARC A:SAVE.ARK *.*  ; List all files in CP/M archive SAVE on drive A
B>UNARC A:SAVE.ARC *.*  ; List all files in MS-DOS archive SAVE on drive A
A>UNARC SAVE            ; Same as either of above
A>UNARC SAVE *.* N      ; Same as above (no screen pauses)
A>UNARC SAVE *.DOC      ; List just .DOC files
A>UNARC SAVE READ.ME    ; Typeout the file READ.ME
A>UNARC SAVE READ.ME N  ; Typeout the file READ.ME (no screen pauses)
A>UNARC SAVE A:         ; Extract all files to drive A
A>UNARC SAVE B:*.DOC    ; Extract .DOC files to drive B
A>UNARC SAVE C:READ.ME  ; Extract file READ.ME to drive C
A>UNARC SAVE PRN.DAT P  ; Print the file PRN.DAT (no formatting)
A>UNARC SAVE *.* C      ; Check validity of all files in archive

Copyright (C) 1986, 1987 by Robert A. Freed
```

Wizytówki i spisy komend programów ARKP i VNARC

```
Archive File = TEST.ARK

Name          Length  Disk  Method   Ver Stored Saved   Date     Time    CRC
============ ======= ==== ======== === ======= ===== ========= ====== ====
LETTER1 .TXT    1792   2k  Crunched  8    1173  35%  13 May 91 12:01a  5C34
LETTER2 .TXT    2432   3k  Crunched  8    1622  34%  13 May 91 12:01a  B9B9
LETTER3 .TXT    1920   2k  Crunched  8    1266  35%  13 May 91 12:01a  F2E1
LETTER4 .TXT     896   1k  Crunched  8     606  33%  13 May 91 12:02a  876F
LETTER5 .TXT    1920   2k  Crunched  8    1244  36%  13 May 91 12:02a  D30A
LETTER6 .TXT    1536   2k  Crunched  8    1071  31%  13 May 91 12:02a  3D04
LETTER7 .TXT    1664   2k  Crunched  8    1081  36%  13 May 91 12:02a  EAE5
TASBBIT .BIN     384   1k  Crunched  8     262  32%  13 May 91 12:02a  1B91
TASCODE1.BIN   14208  14k  Crunched  8   12909  10%  13 May 91 12:02a  30A9
TASCODE2.BIN   16256  16k   Packed   3   14110  14%  13 May 91 12:04a  E9C3
TASCODE3.BIN    1536   2k   Packed   3    1499   3%  13 May 91 12:10a  CD3B
TASWORD .BAS    2048   2k  Crunched  8    1569  24%  13 May 91 12:10a  CE50
        ====  ======= ====               ======= ===                   ====
Total     12    46592  49k                 38412  18%                   5C88
```

UNARC w działaniu

```
A>arkp test let*.* tas*.*
ARK version 0.4
Brian E. Moore
CP/M Plus version
Creating new archive: TEST.ARK
Adding file: LETTER1.TXT analyzing,  crunching, done.
Adding file: LETTER2.TXT analyzing,  crunching, done.
Adding file: LETTER3.TXT analyzing,  crunching, done.
Adding file: LETTER4.TXT analyzing,  crunching, done.
Adding file: LETTER5.TXT analyzing,  crunching, done.
Adding file: LETTER6.TXT analyzing,  crunching, done.
Adding file: LETTER7.TXT analyzing,  crunching, done.
Adding file: TASBBIT.BIN analyzing,  crunching, done.
Adding file: TASCODE1.BIN analyzing,  crunching, done.
Adding file: TASCODE2.BIN analyzing,  packing, done.
Adding file: TASCODE3.BIN analyzing,  packing, done.
Adding file: TASWORD.BAS analyzing,  crunching, done.
```

Przykład działania programu ARKP

UNZIP

Trzecim programem, o którym dziś piszę, jest UNZIP. Został on stworzony do jednego celu: rozpakowywania archiwów stworzonych przez program PKZIP (lub kompatybilny) na IBM PC. Program ten jest bardzo uproszczony i potrafi wykonać tylko dwie czynności: rozpakowanie całego archiwum oraz sprawdzenie poprawności wszystkich plików.

Można go wywołać na dwa sposoby:
UNZIP archiwum lub **UNZIP archiwum E**
Pierwszy sposób powoduje sprawdzenie poprawności, drugi rozpakowanie. UNZIP umieszcza rozpakowane pliki na aktualnym dysku — nie można tego zmienić.

POWOLI I SPOKOJNIE

Te słowa dosyć dokładnie opisują szybkość działania wszystkich opisanych programów. W porównaniu z ich odpowiednikami na IBM PC są one koszmarnie wolne — ale nie jest to wina algorytmu. Powolność jest częściowo spowodowana różnicą w szybkości komputerów, częściowo zaś — małą ilością dostępnej pamięci. Przy braku pamięci konieczne jest przechowywanie pośrednich wyników na dysku, co dosyć skutecznie zwalnia pracę. Poniższa tabelka pokazuje czasy kompresji i dekompresji archiwum składającego się z kilku plików tekstowych i binarnych:

PROGRAM	CZYNNOŚĆ	CZAS
UNARC	rozpakowanie	1 min 44 sek
ARC	spakowanie	7 min 10 sek
UNZIP	rozpakowanie	5 min 27 sek
PAK (MSDOS)	spakowanie	27 sek
PAK (MSDOS)	rozpakowanie	30 sek

Powolność UNZIP-a jest dość nieoczekiwana, szczególnie w sytuacji, gdy programy PKZIP/PKUNZIP (na IBM PC) zaliczają się do najszybszych archiwerów. Podobnie trudna do wyjaśnienia jest różnica czasu pracy programów ARC i UNARC — pecetowy PAK pakuje i rozpakowuje z jednakową szybkością (trzysekundowa różnica może być wyjaśniona „śmietnikiem" na dysku i wynikającym z tego spowolnieniem zapisu), jednak powodem jest prawdopodobnie intensywne korzystanie przez ARC-a z dysku (przechowywanie pośrednich postaci danych)

Michał Szokoło

[1] Archiwizacja nie zawsze jest równoznaczna kompresji. Można archiwizować dane bez kompresowania ich, np. łącząc kilka plików w jeden duży (co i tak daje pewną oszczędność).

dBMAN



Ostatnio w „Bajtku" i „Moim Atari" wiele miejsca poświęcamy na opisy różnego rodzaju emulatorów, które w lepszy lub gorszy sposób zamieniają ST w IBM, lecz zawsze pracują dosyć wolno.

Najczęściej jednak taka zamiana jest zupełnie zbędna. Większość programów użytkowych dla komputerów IBM posiada bowiem swoje odpowiedniki dla Atari ST. Jednym z takich programów jest opisywany w „Bajtku" 4/91 **LDW POWER**, który bardzo dobrze zastępuje **Lotus 1-2-3** i może korzystać z jego danych.

Wśród programów obsługujących bazy danych takim standardem IBM-owskim jest **dBASE**. Aby korzystać z tego programu na Atari ST, wcale nie trzeba stosować emulatora. Skutecznie zastąpi go program **dBMAN** firmy VersaSoft, który stanowi wierną implementację **dBASE II** na Atari ST. Program ten działa na każdym komputerze ST, który posiada co najmniej 1 MB pamięci RAM (1040ST), a także na 520ST z systemem operacyjnym znajdującym się w pamięci ROM. Praktycznie działa więc na prawie wszystkich ST.

Program umieszczony jest na dwóch dyskietkach 3,5". Pierwsza z nich zawiera właściwy **dBMAN** oraz pomocnicze pliki z komentarzami, opisami i dodatkowymi informacjami. Na drugiej dyskietce znajduje się natomiast pełny podręcznik do nauki obsługi programu wywoływany poprzez **dBMAN** poleceniem DO TUTOR. Oczywiście, legalny nabywca programu dysponuje również instrukcją wydrukowaną w postaci książki.

W trakcie pracy nie trzeba jednak korzystać z książki, w każdej chwili można bowiem skorzystać z wbudowanej funkcji HELP. Wymaga ona jedynie skopiowania na dyskietkę zawierającą dane trzech niewielkich plików z komentarzami opisującymi działanie poszczególnych funkcji. Oryginalny program **dBASE** posiada ponadto funkcję ASSIST, która powoduje wyświetlanie wiersza z poleceniami, co znacznie ułatwia korzystanie z programu początkującym użytkownikom. Niestety, **dBMAN 2.01** funkcją tą nie dysponuje — została ona dopiero dodana do programu w wersji **3.00**.

Ponieważ nie wszyscy posiadacze ST orientują się w możliwościach programów przeznaczonych dla komputerów IBM, to wypada choćby pobieżnie przedstawić możliwości **dBMANa**. Trudno tu jednak mówić o maksymalnej pojemności bazy, gdyż jest ona ograniczona w zasadzie wyłącznie pojemnością posiadanych nośników (dyskietki elastyczne lub dysk twardy) oraz rozmiarem pamięci RAM pozostawionej do dyspozycji programu. W każdym przypadku można wszakże pracować równocześnie na dziesięciu różnych bazach zarówno niezależnie, jak i dokonując operacji przesłań między poszczególnymi bazami ich dowolnych elementów (struktur, pól, rekordów, wyrażeń itd.).

dBMAN 2.01 rozpoznaje 76 różnych poleceń, z których większość może występować w kilku odmianach dotyczących zupełnie różnych części i funkcji bazy. Na przykład, polecenie MODIFY może dotyczyć struktury bazy danych (MODIFY STRUCTURE), nazwy pola w bazie (MODIFY FIELDNAME) lub raportu generowanego z bazy (MODIFY REPORT), a w wersji **3.00** także pliku poleceń (MODIFY COMMAND). Uzupełnieniem tych poleceń są liczne funkcje: znakowe (28), logiczne (8) i arytmetyczne (26) oraz dwie funkcje daty i dwie funkcje menu. Wersja **3.00** została rozbudowana o 5 nowych poleceń i jedną funkcję menu.

Wśród poleceń godnych wyróżnienia występują między innymi instrukcje strukturalne i warunkowe CASE, IF/ELSE i WHILE. Są one szczególnie użyteczne podczas tworzenia plików poleceń, które realizują bardziej złożone zadania (patrz niżej). Bardzo pożyteczne jest również polecenie COPY, które pozwala na wybiórcze (według zadanych warunków) kopiowanie rekordów i struktur między różnymi bazami. Dla osób korzystających z komputerów IBM cenne okaże się z pewnością polecenie CONVERT służące do przekształcenia bazy danych stworzonej przez **dBASE** na format **dBMANa**.

Podobnie, bogactwo funkcji zaimplementowanych w programie może zadowolić najbardziej wymagającego użytkownika. Dla zaprezentowania tego bogactwa wystarczy wymienić kilka z nich. Na przykład, DTOK zamienia datę na format „rok/miesiąc/dzień", co pozwala na proste sortowanie według tego parametru. Natomiast funkcja DOW, której parametrem jest data, zwraca w wyniku liczbę (od 1 do 7) określającą dzień tygodnia. Najciekawiej wypada jednak przegląd funkcji logicznych, które między innymi umożliwiają sprawdzenie istnienia pliku na dyskietce i włączenia drukarki.

Jedną z najbardziej użytecznych operacji powodującą, iż korzystanie z **dBMANa** jest szybkie i proste, jest możliwość programowania operacji, które będą później wykonywane automatycznie. Realizuje się to przez zapisanie sekwencji poleceń w oddzielnym pliku, a następnie wywoływanie ich poleceniem DO. Plik taki może się kończyć powrotem do właściwego programu lub przejściem do następnego pliku poleceń, dzięki czemu możliwe jest stworzenie biblioteki krótkich programów realizujących proste zadania, a potem konstruowanie z nich dłuższych ciągów programów wykonujących nawet bardzo złożone zadania. Uniwersalność takich programów można uzyskać przez wykorzystanie polecenia INPUT, funkcji INKEY oraz wymienionych wcześniej instrukcji warunkowych.

Dla zupełnie początkujących użytkowników korzystanie z **dBMANa** (szczególnie w wersji 2.01) z pewnością nie będzie łatwe, lecz po nabraniu wprawy docenią oni to, o czym wie każdy bardziej doświadczony użytkownik: dobry program potrafi znacznie ułatwić pracę i zaoszczędzić wiele czasu. Bez żadnych wątpliwości **dBMAN** może być zaliczony do takich właśnie dobrych programów.

Wojciech Zientara

Wygląd ekranu z programem dBMAN po wywołaniu listy dostępnych poleceń.

```
CMD:


MSG:                                          FP              Rec
                                  F7=Mem F8=Sta F9=Err F10=Cmd

At the "CMD:" prompt, type any of these names and press the "F10" key:

 ?        COMPARE   EDIT      IF        MODINDEX   RENAME    SORT
 ??       CONTINUE  EJECT     IGNORE    NOTE       REPLACE   STORE
 @        CONVERT   ELSE      INC       OTHERWISE  REPORT    SUM
 ACCEPT   COPY      ENDCASE   INDEX     PACK       RESTORE   TEXT
 APPEND   COUNT     ENDDO     INPUT     PAUSE      RETRY     TOTAL
 ASSIGN   COPYREC   ENDIF     JOIN      PRINT      RETURN    UPDATE
 BEEP     CREATE    ERASE     LIST      QUIT       SAVE      USE
 CANCEL   DEC       EXIT      LOCATE    READ       SEEK      UNINDEX
 CASE     DELETE    FIND      LOOP      RECALL     SELECT    WAIT
 CLEAR    DISPLAY   FORMFEED  MERGE     REINDEX    SET       ZAP
 CLOSE    DO        GO        MODIFY    RELEASE    SKIP


^D » Keys, Syntax, Functions @ Esc » CMD:
```

ULEPSZANIE ST

Posiadamy w redakcji komputer Atari 520STM. Jest to „stare" urządzenie o małej pamięci i stosunkowo niewielkich możliwościach. Zamiast jednak zamieniać je na nowszy i lepszy model, zainspirowani artykułem „Staruszek ST" (Moje Atari 3), postanowiliśmy sprawdzić, co jeszcze można z niego wycisnąć. Okazało się, że dużo.

Pamięć

Najważniejszą modernizacją jest powiększenie pojemności pamięci. Niemal wszystkie programy użytkowe wymagają już jednego megabajta RAM. Nasza pięćsetdwudziestka coraz częściej meldowała więc „Not enough free memory". Jak szaleć, to szaleć; nie bawiliśmy się w drobiazgi i zainstalowaliśmy od razu pełne 4 MB. Nie obyło się jednak bez problemów.
Już wcześniej został w naszym ST zainstalowany system PC-Speed (Bajtek 5/91). Wewnątrz obudowy pozostało zatem niewiele miejsca. Jednak firma **TOMS** (nie bez trudu) wymontowała stare układy pamięci i „upchnęła" w ich miejsce 32 nowe układy o organizacji 1048576 x 1 bit. Daje to w rezultacie 4194304 bajty (lub 4096 KB) pamięci. Taki też rozmiar podaje program RAMTEST.PRG służący do sprawdzania pamięci komputera.

Teraz możemy już „poszaleć". W komputerze mieszczą się bez problemu nawet bardzo duże rysunki, skomplikowane, wielostronicowe dokumenty Calamusa, a często jeszcze część pamięci jest wykorzystywana na liczne akcesoria i ramdysk o pojemności 1 MB. Zagospodarowanie tak dużej przestrzeni adresowej nie sprawia komputerowi żadnego kłopotu. Jedynym zewnętrznym objawem pozwalającym dostrzec zmianę jest nieco dłuższy czas startowania systemu, co wynika z konieczności sprawdzenia, ile pamięci jest do dyspozycji i czy jest ona sprawna.

Boot selector

Jednym z nowych urządzeń zewnętrznych współpracujących z naszym ST jest stacja dla dyskietek o średnicy 5,25 cala (zob. artykuł „Stacja 5,25" do ST"). Często zachodzi potrzeba uruchomienia systemu z dyskietki 5,25", a więc ze stacji B zamiast z A. Istnieją wprawdzie programy, które umożliwiają wybór stacji do odczytu wstępnego, lecz muszą one być wczytywane ze stacji A. Znacznie prostszym rozwiązaniem jest zainstalowanie przełącznika, który umożliwia wybór

stacji. Przełącznik taki (umieszczony na tylnej ściance komputera) ma dwa położenia. W położeniu „Normal" komputer zachowuje się tak, jakby żadnej przeróbki nie było. Natomiast po przestawieniu przełącznika w położenie „Inverse" wstępny odczyt *(booting)* i inicjowanie systemu są przeprowadzane ze stacji B.

Ktoś może powiedzieć, że jest to „bajer" W pewnym sensie będzie miał rację. Jednak koszt takiej przeróbki jest minimalny, zaś korzyść niewspółmiernie duża — oczywiście, jeśli posiada się stację pięciocalową.

Stereo

Prawdziwym „bajerem" jest za to kolejny drobiazg, który zmieścił się w naszej maszynce. Pozazdrościliśmy bowiem kolegom „amigantom" stereofonicznego dźwięku i wyposażyliśmy nasz ST w prostą modyfikację, która coś takiego symuluje.

Na tylnej ściance komputera znalazły się dwa gniazda CINCH. Są do nich doprowadzone sygnały lewego i prawego kanału stereo. Poziom tych sygnałów wynosi kilkaset miliwoltów, można więc połączyć je bezpośrednio z wejściem dowolnego wzmacniacza stereofonicznego. Działanie tego urządzenia jest oparte na wewnętrznej budowie komputera, który posiada trzy niezależne generatory dźwięku. Dźwięk tworzony przez pierwszy generator jest przesyłany do jednego kanału stereo, z drugiego generatora — do drugiego kanału, a z trzeciego — do obu kanałów.

Takie prościutkie rozwiązanie daje doskonałe efekty, jeżeli dźwięk jest generowany przez komputer, na przykład w grach. Niestety, nie nadaje się ono do odtwarzania dźwięku zdigitalizowanego, gdyż różne tony tych samych instrumentów słychać z różnych kanałów. W takim wypadku konieczne jest odłączenie stereofonii dodatkowym przełącznikiem, który znajduje się obok gniazd.

Podsumowanie

Nawet stary komputer można „odmłodzić", czego przykłady są pokazane wyżej. Pojawia się jednak pytanie o przydatność takich rozszerzeń. Odpowiedź zależy przede wszystkim od przeznaczenia komputera. Z pewnością powiększenie rozmiaru pamięci przyda się każdemu. Niejako przy okazji posiadacze stacji 5,25" powinni również zainstalować sobie przełącznik stacji wstępnego odczytu, namiętni (i muzykalni) gracze będą zaś zachwyceni nowymi efektami dźwiękowymi swoich ulubionych gier.

Opisane tu modyfikacje Atari ST stanowią jedynie niewielki ułamek możliwości. Takie zostały nam zaproponowane przez firmę **TOMS**. Jeżeli inne firmy zmieszczą coś jeszcze do naszego komputera, niewątpliwie poinformujemy Czytelników, czy warto to zainstalować i jakie są korzyści (oraz utrudnienia) w pracy i zabawie.

Marek Zachar

Producent: TOMS
Tadeusz Okoń
Marek Stolarski
Warszawa, ul. Widok 14/1
tel. (0-22) 27-16-01
(0-2) 641-54-29





STACJA 5,25" DO ST

W komputerach Atari ST standardowo stosowane są stacje dysków przeznaczone dla dyskietek o średnicy 3,5 cala. Dyskietki te mają niewielkie rozmiary i twardą kopertę, są jednak droższe od dyskietek 5,25 cala. Większość komputerów IBM oraz niemal wszystkie 8-bitowe są wyposażone w stacje dla dyskietek 5,25 cala. Wywołuje to problemy przy przenoszeniu danych i programów pomiędzy tymi komputerami. Jednym z rozwiązań jest zastosowanie stacji dla dyskietek 5,25 cala współpracującej z Atari ST.

Jednym z producentów takich stacji jest Przedsiębiorstwo „FORMAT". Stacja **FORMAT** jest dwustronną (dwugłowicową) stacją dysków elastycznych 5,25 cala przeznaczoną do współpracy z 16-bitowymi komputerami Atari ST. Umożliwia ona jedno lub dwustronny zapis i odczyt dyskietek w czterech różnych formatach. Przyłączenie do komputera jest zrealizowane poprzez standardowe złącze szeregowe FLOPPY DISK Atari ST.

BUDOWA

Obudowa stacji jest wykonana ze stalowej blachy pomalowanej na szaro dla uzyskania kolorystycznej zgodności z resztą zestawu. Przednia ścianka obudowy jest utworzona przez płytę czołową napędu. Na tylnej ściance znajduje się wyłącznik zasilania oraz przełączniki stron i liczby ścieżek. Ponadto wyprowadzone są tam dwa przewody: zasilający i łączący z komputerem. Dzięki umieszczeniu z tyłu wszystkich przełączników uzyskano niewielkie rozmiary stacji. Wynoszą one: szerokość 147 mm, wysokość 41 mm (razem z nóżkami 49 mm) i długość 317 mm.

Wewnątrz stacji znajduje się również zasilacz, co jest już standardowym rozwiązaniem w sprzęcie tej klasy. Miejsce na jego umieszczenie wygospodarowano przez zwiększenie długości obudowy. Dlatego właśnie jest ona znacznie dłuższa od typowych stacji Atari ST.

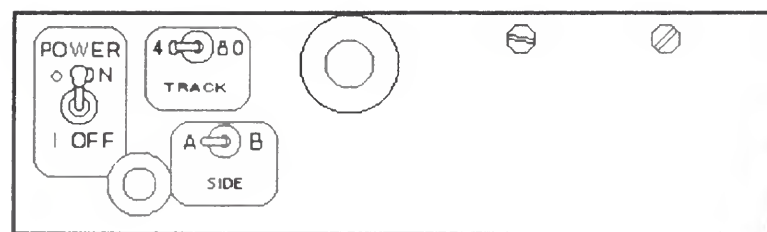
PRACA

Stacja **FORMAT** rozpoznaje cztery formaty dyskietek. Są one następujące:

SS40 — jednostronny format IBM S-9 (1 strona, 40 ścieżek po 9 sektorów po 512 bajtów = 180 KB),
SS80 — jednostronny format ST (1 strona, 80 ścieżek po 9 sektorów po 512 bajtów = 360 KB),
DS40 — dwustronny format IBM D-9 (2 strony, 40 ścieżek po 9 sektorów po 512 bajtów = 360 KB),
DS80 — dwustronny format ST (2 strony, 80 ścieżek po 9 sektorów po 512 bajtów = 720 KB),

Istnieje ponadto możliwość pracy z dyskietkami o większej pojemności (np. 900 KB) poprzez dodanie przy formatowaniu dodatkowych sektorów i ścieżek. Postępowanie takie nie jest jednak zalecane przez producenta ze względu na możliwość wystąpienia błędów przy odczycie i zapisie.

Wybór formatu, w którym pracuje stacja, jest dokonywany za pomocą przełącznika określającego liczbę ścieżek na dyskietce (40 lub 80). Drugi przełącznik umożliwia zapis i odczyt na drugiej stronie dyskietki zapisanej w formacie jednostronnym bez jej odwracania. Eliminuje to konieczność przekładania dyskietek, które mają dwie strony zapisane w formatach jednostronnych. Przy ustawieniu tego przełącznika do pracy na pierwszej stronie dyskietki stacja samoczynnie rozpoznaje, czy dyskietka jest zapisana jednostronnie, czy dwustronnie.



EKSPLOATACJA

Obsługa stacji **FORMAT** jest bardzo prosta, lecz mimo to ograniczenie się do instrukcji zajmującej jedną stronę formatu A5 wydaje się zbytnią oszczędnością. Pewne utrudnienia powodowane są przez umieszczenie przełączników z tyłu stacji. Nie widać z tego powodu ich położenia, co kilkakrotnie spowodowało nieprawidłowe odczytywanie dyskietek. Początkowo miałem także kłopoty ze zlokalizowaniem przełączników bez wstawania z krzesła i zaglądania za stację. Uwagi te dotyczą w szczególności przełącznika stron, który eliminuje uciążliwe przekładanie dyskietek na drugą stronę. Aby zachować niewielkie wymiary stacji, kłopot ten mógłby być również rozwiązany przez umieszczenie na przedniej ściance stacji diod świecących, które sygnalizowałyby aktualny stan przełączników.

Instrukcja nie wspomina ani słowem o tym, czy położenie przełączników można zmieniać, gdy stacja jest włączona. Wprawdzie operacje takie nie spowodowały uszkodzenia stacji, lecz nie wiadomo, czy był to przypadek (miałem szczęście?), czy też zasada. Jest to szczególnie istotne przy kopiowaniu dyskietek IBM na format 80-ścieżkowy oraz przy pracy z dyskietkami, z których część jest zapisana jednostronnie a część dwustronnie.

Podstawową korzyścią z użytkowania stacji **FORMAT** jest uproszczenie z korzystania oprogramowania przeznaczonego dla komputerów IBM. Testowany w „Moim Atari 5" emulator **PC-Ditto** umożliwia na taką okazję skonfigurowanie systemu w ten sposób, aby stacja 5,25" była rozpoznawana jako A, zaś stacja 3,5" jako B (czyli odwrotnie). Dzięki temu całe oprogramowanie IBM można przechowywać na dyskietkach 5,25", stację 3,5" zaś używać do przenoszenia danych z i na dyskietki Atari ST. Jest to o tyle cenna zaleta, że większość programów użytkowych stosuje identyczny lub bardzo podobny sposób zapisu informacji na dyskietce.

Pewne problemy pojawiają się czasem przy korzystaniu z formatu 40-ścieżkowego. Jest to spowodowane tym, iż stacja **FORMAT** tylko „udaje" stację 40-ścieżkową, lecz pozostaje nadal 80-ścieżkową. Bliższe informacje na ten temat można znaleźć w artykule „Formatowanie dyskietek" opublikowanym w „Bajtku" 3/91 (klan IBM). Instrukcja „załatwia" tą sprawę jednym zdaniem: „Mogą wystąpić problemy podobne jak w PC AT przy **pracy** ze stacją 1,2 MB w formacie 360 KB"

PODSUMOWANIE

Stacja **FORMAT** będzie z pewnością cennym nabytkiem dla każdego posiadacza Atari ST. Największy pożytek będą jednak mieli z niej ci użytkownicy, którzy często korzystają z komputerów IBM oraz przeznaczonego dla nich oprogramowania. Powinny się nią zainteresować również te osoby, które posiadają (lub niedawno posiadały) Atari XL/XE i chciałyby przenieść zbiory danych unikając ich żmudnego przepisywania.

Marek Zachar

Zalety:

- wbudowany zasilacz
- niewielkie rozmiary
- emulacja stacji 40-ścieżkowej
- możliwość korzystania z dwóch stron bez przekładania dyskietki sformatowanej jednostronnie

Wady:

- umieszczenie przełączników z tyłu i brak sygnalizacji ich stanu
- nadmiernie uproszczona instrukcja

Producent:
Przedsiębiorstwo „FORMAT"
00-502 Warszawa, ul. Bracka 4 p. 7
tel. 29-60-47 wewn. 25

BASIC BOOTSTRAP

Każdy posiadacz stacji dysków miał chyba do czynienia z inicjalizerami do programów napisanych w języku maszynowym. Ułatwiają one w znacznym stopniu uruchamianie tych programów. Dla programów w BASIC-u jest już o taki inicjalizer znacznie trudniej, a te, które są, nie zawsze spełniają swoje zadania. Z tych właśnie powodów powstał BASIC BOOTSTRAP.

Właściwy program jest przedstawiony na listingu 1. Jest on napisany w Atari BASIC-u. Zajmuje niewiele ponad 2 KB pamięci. Program ten po uruchomieniu, wyświetla nazwy wszystkich plików znajdujących się na dyskietce, z wyjątkiem tych, które mają rozszerzenia: .SYS, .COM, .EXE, .SMS oraz tych, które są niezabezpieczone. Pozwala to na uniknięcie wyświetlania nazw plików, które nie powinny się pojawiać, w tym wszelkich plików systemowych (takich jak DOS, AUTORUN.SYS itp.). Na przykład, jeśli dany program zajmuje kilka plików, to wystarczy zabezpieczyć tylko plik zawierający część ładującą, a pozostałe wykorzystywane przez ten program pozostawić niezabezpieczone. Unikniemy w ten sposób pojawiania się w inicjalizerze zbędnych nazw (bo i tak pliki te nie mogą być uruchomione samodzielnie).

Po wyświetleniu katalogu wystarczy wcisnąć znak stojący przy nazwie programu, który chcemy uruchomić i czekać, aż zostanie on wczytany i uruchomiony.

LISTING 1

```
UB   0 REM BASIC BOOTSTRAP
BU   1 REM FOR ALL BASIC FILES
AD   2 REM Krzysztof Klimczak
DV   3 REM (c) 1991, Sp. Bajtek
NJ   4 REM
TE   10 DIM FIL$(700),A$(20),B$(3):POKE 566
     ,158
ME   20 GOSUB 1000
QG   22 Q=1:Q1=0
MY   25 GOSUB 2000
WU   30 IF WSK OR Q=17 THEN POSITION 12,23:
     ? "<RETURN> MORE";
CO   40 POSITION 3,21:? "Select program: ";
     :CLOSE #1:OPEN #1,4,0,"K":POKE 694,0:P
     OKE 702,64
IA   50 GET #1,K
KQ   51 IF K=27 THEN RUN
MZ   52 IF K<65 OR K>ZN-1 AND K<>155 THEN 5
     0
QG   55 IF K=155 AND  NOT WSK THEN 50
IM   60 IF K=155 AND Q=1 THEN Q=16:Q1=32:GO
     TO 25
AC   65 IF K=155 THEN 22
SY   67 TRAP 80
FY   70 POKE 752,1:? CHR$(K):K=K-65:A$="":A
     $(1,2)="D:":A$(3,13)=FIL$(K*11+1,K*11+
     11)
AD   72 IF WSK THEN A$(3,13)=FIL$((K+32)*11
     +1,(K+33)*11)
RM   73 B$(1,3)=A$(11,13)
LM   74 FOR T=3 TO 10:IF A$(T,T)=" " THEN P
     OP :GOTO 76
NA   75 NEXT T
HP   76 A$(T,T)=".":A$(T+1,T+3)=B$(1,3)
QJ   77 POKE 566,146:POKE 82,2:RUN A$
YS   80 GRAPHICS 0:? "Nie moge uruchomic te
     go pliku..":FOR T=1 TO 300:NEXT T:RUN
AZ   99 STOP
DT   1000 IL=0:TRAP 1030:CLOSE #1:OPEN #1,6
     ,0,"D:*.*":FOR T=0 TO 63:INPUT #1,A$:B
     $=A$(11,13)
LI   1002 IF A$(1,1)<>"*" THEN 1020
EE   1005 IF B$="SYS" OR B$="COM" OR B$="EX
     E" OR B$="SMS" THEN 1020
VW   1010 FIL$(IL*11+1,IL*11+11)=A$(3,13):I
     L=IL+1
JM   1020 NEXT T
NR   1030 WSK=0:IL=IL-1:IF IL>31 THEN WSK=1
AL   1040 RETURN
BP   2000 GRAPHICS 0:POKE 82,3:? :? CHR$(12
     5):? "BASIC BOOTSTRAP  by   K. Klimcza
     k"
BQ   2005 IF IL=-1 THEN ? :? :? :? "   Brak
      plikow. Zmien dyskietke.":? CHR$(127)
     ;"     Wcisnij ESC..":ZN=64:GOTO 2070
ZN   2010 ? :? :ZN=65:IF IL<16+Q1 THEN FOR
     T=Q1 TO IL:? CHR$(ZN);" - ";FIL$(T*11+
     1,T*11+11):ZN=ZN+1:NEXT T:GOTO 2070
ZM   2020 FOR T=Q1 TO Q1+15:? CHR$(ZN);" -
     ";FIL$(T*11+1,T*11+11):ZN=ZN+1:NEXT T
FH   2025 POSITION 20,4
IA   2030 IF IL<32+Q1 THEN FOR T=16+Q1 TO I
     L:POKE 85,20:? CHR$(ZN);" - ";FIL$(T*1
     1+1,T*11+11):ZN=ZN+1:NEXT T:GOTO 2070
SJ   2040 FOR T=Q1+16 TO Q1+31:POKE 85,20:?
      CHR$(ZN);" - ";FIL$(T*11+1,T*11+11):Z
     N=ZN+1:NEXT T
AV   2070 RETURN
VF   2080 REM ***
```

Jeżeli w czasie ładowania wystąpi błąd, program poinformuje o tym i uruchomi się od nowa (najczęściej błąd ten jest spowodowany próbą uruchomienia programu, który nie jest napisany w BASIC-u). Jeśli na dyskietce znajduje się więcej niż 32 pliki do uruchomienia, to komputer wyświetli pierwsze 32 i będzie oczekiwał na wciśnięcie klawisza. Gdy tym klawiszem będzie ‹RETURN›, to zostanie wyświetlona dalsza część plików z katalogu dyskietki. W ten sposób program może załadować każdy plik znajdujący się na dyskietce.

Na koniec opisu jeszcze jedna pomocna funkcja: jeśli w czasie oczekiwania na wybór programu wciśniemy ‹ESC›, to program uruchomi się od początku odczytując katalog dyskietki, która będzie się aktualnie znajdowała w stacji dysków. Umożliwia to szybkie przeglądanie i ładowanie programów w BASIC-u z innych dyskietek.

W celu uruchomienia programu należy najpierw wpisać uważnie program z listingu 1. Po jego przepisaniu nagrywamy go na dyskietkę instrukcją SAVE pod nazwą:

- AUTORUN.BAS, gdy chcemy wczytywać programy napisane w Turbo BASIC-u. Oczywiście na tej dyskietce musi się znajdować jeszcze DOS.SYS i sam Turbo BASIC nagrany pod nazwą AUTORUN.SYS.
- INIT.SMS, gdy chcemy wczytywać programy napisane w Atari BASIC.

W tym drugim przypadku musimy jeszcze przepisać i uruchomić program z listingu 2. Generuje on plik AUTORUN.SYS, który umożliwi automatyczne uruchomienie inicjalizera.

Na żądanie włożenia dyskietki wkładamy tą, na której nagraliśmy plik INIT.SMS i wciskamy ‹RETURN›. Na dyskietce tej musi się również znajdować plik DOS.SYS, który zapisuje się funkcją „H" każdego DOS-u (zapisywany przy tym plik DUP.SYS można skasować).

Czynności te mogą być wykonane zarówno na czystej (uprzednio sformatowanej) dyskietce jak i na dyskietce zawierającej już pliki, które mają być uruchamiane.

UWAGA: należy przestrzegać nazw pod którymi mają być zapisane pliki programu BASIC BOOTSTRAP.

Teraz już można włożyć dyskietkę do stacji, włączyć komputer i czekać na uruchomienie inicjalizera. Komputer włączamy z wciśniętym klawiszem ‹OPTION›, gdy dyskietka zawiera Turbo BASIC i BEZ ‹OPTION›, gdy programy na niej są napisane w Atari BASIC-u.

W celu skopiowania inicjalizera na inną dyskietkę wystarczy tylko skopiować pliki INIT.SMS i AUTORUN.SYS (lub AUTORUN.BAS).

Krzysztof Klimczak

LISTING 2

```
SV   0 REM GENERATOR AUTORUN.SYS
QC   1 REM uruchamiajacego INIT.SMS
AD   2 REM Krzysztof Klimczak
DV   3 REM (c) 1991, Sp. Bajtek
NJ   4 REM
VA   10 DIM AUT$(140)
QM   20 RESTORE 100:FOR T=0 TO 6:S=0:FOR T1
     =1 TO 20:READ A:S=S+A:AUT$(T*20+T1)=CH
     R$(A):NEXT T1:READ SUMA
RV   30 IF SUMA<>S THEN ? "Blad danych w wi
     erszu 10";T:? :? :LIST T+100:STOP
MN   40 NEXT T
GG   50 ? :? "O.K. Wloz dyskietke. Wcisnij
     RETURN..":POKE 764,255
SS   60 IF PEEK(764)=255 THEN 60
VA   70 IF PEEK(764)<>12 THEN 60
TP   80 TRAP 90:OPEN #1,8,0,"D:AUTORUN.SYS"
     :? #1;AUT$(1,139);:CLOSE #1:? :? "WSZY
     STKO W PORZADKU !!":END
RC   90 ? :? "BLAD W CZASIE ZAPISU. SPRAWDZ
      DYSK     I URUCHOM JESZCZE RAZ":STOP
DO   100 DATA 255,255,160,6,162,6,76,175,6,
     175,6,251,6,160,11,185,0,228,153,163,2
     439
ZV   101 DATA 6,136,16,247,169,222,141,167,
     6,169,6,141,168,6,172,170,6,174,169,6,
     2297
TP   102 DATA 232,208,1,200,142,246,6,140,2
     47,6,169,163,141,33,3,169,6,141,34,3,2
     290
SK   103 DATA 96,172,0,6,208,10,169,0,141,3
     3,3,169,228,141,34,3,185,1,6,206,1811
NA   104 DATA 0,6,72,32,251,6,104,160,1,96,
     253,6,255,6,108,250,191,68,2,68,1935
HK   105 DATA 2,0,9,0,9,0,1,226,2,227,2,160
     ,6,224,2,225,2,253,6,0,1356
YU   106 DATA 6,15,6,14,155,83,77,83,46,84,
     73,78,73,58,68,34,78,85,82,0,1198
WW   110 REM ***
```

WYCISKANIE WODY —

O kompresji tekstu już kiedyś pisałem — nasłuchałem się wtedy różnych różności o zastosowanym algorytmie — że wcale nie jest zbyt wydajny, że istnieją znacznie lepsze i że właściwie to o nich trzeba było napisać artykuł przeglądowy, a nie o jakichś swoich wymysłach. Ponieważ jednak jestem uparty, postanowiłem jeszcze raz napisać o kompresji przy pomocy wymyślonego przeze mnie, niezbyt wydajnego, za to bardzo eleganckiego algorytmu.

Zanim jednak zajmiemy się samym algorytmem, powiedzmy sobie co nieco o pewnych specjalnych danych — o danych graficznych. Za tym niesłychanie inteligentnie brzmiącym sformułowaniem kryją się wszelkie rodzaje rysunków poczynając od najprostszych, czarno-białych grafik składających się z kilku cienkich kresek, na kolorowo wyrafinowanych obrazach impresjonistów kończąc. Każdy typ rysunku (a można je podzielić właśnie ze względu na liczbę barw i najczęściej występującą formę — kreskę lub plamę) wymaga nieco innych metod traktowania przy obróbce komputerowej.

Ograniczmy nasze rozważania do czerni i bieli, co więcej, pomińmy wszystkie odcienie szarości, jako utrudniające zachowanie binarnego porządku, zgodnie z zasadą jeden piksel — jeden bit. Mamy więc przed sobą obraz, składający się z samych czarnych i białych punktów i jego dotyczyć będą nasze dalsze dywagacje. Co to może być za obraz? Na przykład mapa konturowa, albo wykres, albo rysunek techniczny, albo kartka książki, albo sto innych rzeczy łatwych do odnalezienia w otaczającym nas świecie.

Zapamiętywanie nawet takich, najprostszych rysunków jest niezwykle pamięciożerne. Korzystając z rozdzielczości 300 dpi (*dots per inch,* czyli kropek na cal), będącej w tej chwili pewnym standardem osiągalnym w ręcznych skanerach i drukarkach laserowych, na dokładne zapamiętanie czarno-białego obrazka o formacie jednej strony A4 potrzeba ni mniej ni więcej jak 1087650 bajtów, czyli niemal dokładnie jeden megabajt. Zważywszy, że format A4 jest jednym z mniejszych stosowanych w technice, 300 dpi nie zawsze wystarcza, a kiedy niewielka dokumentacja zawiera kilka lub kilkadziesiąt rysunków, w zastosowaniach praktycznych w grę wchodzą już nie mega, a gigabajty. Przy takich ilościach danych kompresja staje się koniecznością — inaczej nie sposób wyobrazić sobie ich magazynowania lub przesyłania, nawet przy użyciu najszybszych łącz i największych dysków.

Obrazy graficzne, a składające się wyłącznie z punktów dwóch barw w szczególności mają kilka bardzo ciekawych cech. Po pierwsze, zwykle jedna barwa występuje w znacznej przewadze — na rysunku technicznym biały papier zajmuje ponad dziewięćdziesiąt pięć procent powierzchni, a czarne kreski resztę. Po drugie, punkty jednego koloru zwykle występują blisko siebie. Zróbmy pewien eksperyment myślowy — wyobraźcie sobie, że losujemy jeden dowolny punkt z powierzchni rysunku technicznego, po czym sprawdzamy, jakie są kolory sąsiednich punktów. Jeżeli wylosowany punkt jest biały, to zwykle wszystkie otaczające go punkty też będą białe, jeżeli czarny — zwykle przynajmniej jeden z sąsiednich punktów też jest czarny. Można to bardzo ładnie opisać za pomocą rachunku prawdopodobieństwa, jednak nie tym mieliśmy się zajmować.

Podstawowym pomysłem leżącym u podstaw wszystkich metod kompresji jest zastąpienie powtarzalnych fragmentów danych innymi, znacznie od nich krótszymi. W przypadku obrazu czarno-białego z jego cech wynika, że można się na nim spodziewać dużych białych plam (nie mylić z historycznymi). Każdy biały prostokąt możemy zastąpić współrzędnymi jego dwóch wierzchołków — na przykład lewego górnego i prawego dolnego, oraz jakoś zakodowaną informacją o tym, że jest to czysty biały prostokąt. Tym sposobem można zaoszczędzić bardzo dużo — na zapamiętanie białego kwadratu jeden cal na jeden cal (niecałe 6.5 cm^2) potrzeba normalnie 11250 bajtów, jeżeli zapamiętać tylko współrzędne wierzchołków wystarczy osiem bajtów plus kilka na zapamiętanie, że jest to biały obszar. Jest jednak jedno małe ale — śledzenie stanu poszczególnych punktów obrazu (już zakodowany, jeszcze nie zakodowany) bardzo komplikuje potencjalny algorytm opierający się na wyborze jak największych białych prostokątów. Nie tędy więc droga.

Przyznam, że kiedyś sam ugrzązłem w tym właśnie miejscu i zrezygnowałem z dalszych prób wymyślenia algorytmu. Sprawa jednak nie dawała mi spokoju. Kiedy w kilka tygodni później wpadła mi do ręki książka Theo Pavlidisa „Grafika i przetwarzanie obrazów" i znalazłem w niej strukturę danych o nazwie „drzewo czwórkowe", przyszło olśnienie. Drzewo czwórkowe powstaje (z grubsza rzecz biorąc) wtedy, gdy początkowy obraz dzielimy na cztery równe części, następnie każdą z nich znowu na cztery i tak dalej. Jak to z drzewami bywa, nigdzie nie jest powiedziane, że wszystkie gałęzie muszą być tej samej długości — jeżeli w którymś momencie uznamy, że któregoś z podobrazków nie chcemy dzielić dalej na cztery, żadna siła nas do tego nie zmusi.

A kiedy możemy zrezygnować z tego dzielenia? Ano na przykład wtedy, kiedy cały podobrazek jest jednego koloru — i tu pojawia się punkt wspólny pozwalający na połączenie pomysłu kompresji z drzewem czwórkowym. Zamiast wybierać jak największe białe prostokąty i zapamiętywać ich współrzędne, można dzielić systematycznie poddawany kompresji obszar na ćwiartki. Gdy ćwiartka jest całkowicie biała, do zewnętrznego pliku zawierającego poddany kompresji obraz wysyła się zero (binarne), gdy zawiera choć jeden czarny punkt wysyła się jedynkę, po czym ćwiartkę dzieli się na kolejne ćwiartki i operację powtarza się znowu. W pewnym momencie dojdziemy do pojedynczych pikseli — będzie to znak, że dalej już dzielić obrazu nie trzeba. Może się oczywiście zdarzyć, że obszaru nie da się podzielić na cztery części — wtedy, gdy któryś jego rozmiar jest równy jednemu pikselowi. Można wtedy przyjąć, że dwie spośród czterech ćwiartek mają rozmiary zerowe i nie brać ich pod uwagę podczas kompresji. Ustalenie kolejności obrabiania pozostałych w wyniku podziału ćwiartek pozwoli później na odtworzenie oryginalnego obrazu.

Oczywiście od pomysłu do gotowego programu (czy raczej procedury) droga jeszcze daleka, ale przynajmniej widać już konkretny kierunek. Następny etap to próba „ręcznej" realizacji kompresji, pozwalająca na wykrycie ewentualnych niekonsekwencji w projektowanym algorytmie. Taka próba jest zilustrowana na rysunku — nie będę więc dokładnie tłumaczył jej przebiegu, proponuję Wam natomiast prześledzenie kolejnych kroków.

Już gotowy algorytm, zapisany w Pascalu, znajduje się na wydruku. Sama procedura dokonująca kompresji (i jej siostra do dekompresji) jest dzięki zastosowaniu rekurencji niezwykle prosta, trochę pracy wymagało jedynie poprawne ustalenie sposobu tworzenia ćwiartek przekazywanych do dalszej obróbki następnym pokoleniem procedury **scrcompres**. Warto zauważyć, że głębokość rekurencji jest stosunkowo niewielka — zależy bowiem od logarytmu (o podstawie 2) długości dłuższego boku obszaru poddawanego kompresji. Wszystkie pozostałe wykorzystywane w programie procedury i funkcje — **send, get** i **boxempty** — pełnią role pomocnicze. Dwie pierwsze służą do wysyłania i pobierania pojedynczych bitów z pliku zawierającego skompresowany obraz, trzecia sprawdza, czy wewnątrz wskazanego jej obszaru (aktualnej ćwiartki) nie ma jakiegoś czarnego punktu. (Oczywiście pojęcie czarny-biały, zapalony-zgaszony, należy rozumieć elastycznie — na ekranie Herculesa zapalony punkt jest jasny, na wydruku będzie ciemny — w oby przypadkach jest to ten sam „czarny" punkt.)

Jak na razie udało nam się stworzyć pewien algorytm, który powinien (zgodnie ze zdrowym rozsądkiem) dokonywać kompresji obrazów. Ale czy jej rzeczywiście dokonuje? To trzeba było sprawdzić, i to doświadczalnie — generując różne obrazki i poddając je kompresji. Takich prób przeprowadziłem kilkaset (oczywiście nie ręcznie, napisałem w tym celu program). Okazało się, że w zależności od ilości czarnych punktów na obrazie (zawsze poniżej 20%), rozmiar pliku po kompresji był od trzech do dziesięciu razy mniejszy niż rozmiar orygi-

Dokończenie na str. 24

1

11

110

1100

11000

110001

1100011

11000111

110001110

1100011100

11000111001

110001110011

1100011100111

11000111001111

110001110011110

1100011100111100

11000111001111000

Schemat kompresji obrazu zawierającego cztery zapalone piksele. Kolejność rysunków — taka jak przy czytaniu. Grubymi kreskami zaznaczono aktualny podział na ćwiartki (i podćwiartki), na szaro zaznaczona jest ćwiartka badana w danej chwili przez funkcję boxempty. Pod rysunkami znajduje się aktualny stan pliku wyjściowego — ostatnia cyfra dotyczy badanego aktualnie obszaru.

FAX modem DO AMIGI



Od firmy NOGASTA redakcja otrzymała do testowania modem z wbudowanym telefaksem PHONIC SUPREME 9624 wraz z oprogramowaniem przystosowanym do współpracy z komputerem Amiga.

ZESTAW

W skład zestawu wchodzi: modem, zasilacz, przewód telefoniczny (nieprzystosowany do polskich gniazdek), instrukcja obsługi modemu (po angielsku), instrukcja obsługi telefaksu (po niemiecku) oraz dyskietka z oprogramowaniem do telefaksu. Firma Nogasta obiecuje, że już niedługo wszystkie instrukcje będą przetłumaczone na język polski.

Modem podłączany jest za pomocą standardowego przewodu do portu szeregowego komputera (nie musi to być Amiga). Po prawidłowym połączeniu z linią telefoniczną modem jest natychmiast gotowy do pracy.

MODEM

Wchodzący w skład zestawu modem jest zgodny ze standardem Hayes, tak więc nie będzie problemu z jego użytkowaniem — wystarczy kupić odpowiednie oprogramowanie terminala, które nie jest dołączane do zestawu. Modem może pracować z szybkością 300, 1200 oraz 2400 bodów (bitów na sekundę) jednakże bez protokołów wykrywania błędów i kompresji danych (MNP 1-5). Przeciętny użytkownik, będzie zapewne zaskoczony faktem, że jako telefaks urządzenie to może pracować z szybkościami 4800, 7200 i 9600 bodów podczas gdy maksymalną szybkością pracy dla modemu jest prędkość 2400 bodów. Także brak tzw. MNP jest dość bolący, szczególnie ze względu na opłakany stan naszych linii telefonicznych. Pośrednim wyjaśnieniem tego faktu może cena — modemy pracujące z szybkością 9600 kosztują od 400 do 1000 i więcej dolarów, natomiast cena testowanego urządzenia wynosi około 600 DM.

Urządzenie było testowane i pracowało bez problemu przez ponad dwa tygodnie w jednym z warszawskich BBS-ów. Wybieranie numeru jest stosunkowo głośne i charakteryzuje się „pyknięciem" wewnątrz modemu przy każdym impulsie wybierającym. Może to być nocną zmorą dla domowników, którzy śpią w tym samym pokoju, w którym jest zainstalowany modem z komputerem. Z kolei do bezspornych zalet należy możliwość regulacji siły głosu dobywającego się z wewnętrznego głośniczka. Trzeba stwierdzić, że PHONIC SUPREME 9624 jako modem zachowywał się bez problemu.

TELEFAKS

Telefaks jest dostępny dzięki specjalnym instrukcjom rozpoznawanym przez modem (format: „AT+" i polecenie, np. „AT+F1" włącza telefaks). Telefaks będzie współpracował z dowolnym oprogramowaniem wykorzystującym takie polecenia.

Jak już wspomniałem urządzenie komunikuje się z szybkością 4800, 7200 albo 9600 bodów. Niestety, w przypadku Amigi producent oprogramowania obsługującego ostrzega, że transmisja danych z szybkością 9600 jest niemożliwa z powodu domniemanego błędu w ROM i w celu wykorzystania prędkości 9600 należy zakupić odpowiednią kartę zastępująca standardowe złącze szeregowe (RS 232). Z własnego doświadczenia wiem, że na zachodzie modemy 9600 do Amigi to już standard (nie mówiąc o 19200).

OPROGRAMOWANIE

Firma NOGASTA dostarczyła nam dwie dyskietki — jedną oryginalną, dołączaną do zestawu i drugą już „spolszczoną", ale niestety bez polskich liter. Na dyskietce są zapisane dwa programy przeznaczone do obsługi faxu — Receive i MultiFax. Pierwszy z nich służy do odbioru przychodzących telefaksów. Program pozwala zarówno na automatyczną jak i na ręczną obsługę telefaksu (funkcje Auto ON, oraz Pick Up).

Program przetwarza odbierane dane na własny format i zapisuje je w pamięci RAM (RAM-dysk). Jest to konieczne, gdyż stacje dysków nie są w stanie zapisywać danych na dyskietkach z szybkością 9600 bodów. Stosowanie RAM-dysku stwarza jednak pewien problem — wykorzystanie programu na standardowej wersji Amigi 500 z 512 KB RAM jest ograniczone brakiem pamięci. Zastosowanie twardego dysku jest tu rozwiązaniem najlepszym, lecz również znacznie kosztowniejszym od dodatkowej pamięci — po raz kolejny okazuje się, że 1MB RAM dla Amigi to minimum absolutnie niezbędne.

Aby odebrane dane zamienić na format czytelny przez większość edytorów graficznych (IFF) trzeba skorzystać z programu MultiFax. Jest to o tyle niewygodne, że czas potrzebny na wczytanie i uruchomienie programu MultiFax z dyskietki jest stosunkowo długi.

Program MultiFax służy również do nadawania wielostronicowych dokumentów z możliwością uprzedniego przeglądu ich zawartości (opcja PREVIEW) w trzech powiększeniach. Jest to bardzo pomocne, szczególnie gdy przygotowujesz taki dokument „na gorąco". MultiFax komunikuje się z użytkownikiem za pomocą systemu okienek, gadżetów i rozwijanych menu co zdecydowanie ułatwia pracę. Możliwe jest założenie książki telefonicznej zawierającej numery oraz adresy firm lub osób, do których wysyłamy faksy. Wysłanie faksu można zaprogramować na określoną godzinę, np. 5:26 nad ranem. Tu jednak spotkałem się z błędem w programie. Ponieważ korzystam z dysku twardego A590 w moim systemie komórki o adresach 0, 1, 2 i 3 nie zawierają zer tak jak to bywa w „normalnych" Amigach lecz pewne wartości (zależy to od typu Amigi). Gdy opuściłem ustawienie godziny nadania faksu program zaczął oczekiwanie (!) na... wyraz „gdos", zapisany w moim systemie właśnie pod adresem 0. Jest to ewidentne niedociągnięcie programistów, aczkolwiek wystąpienie tego objawu zależy w dużym stopniu od typu Amigi i twardego dysku.

Program oferuje możliwość dołączenia obrazka, który będzie widniał na poszczególnych stronach nadawanego dokumentu; zwykle jest to znak firmowy lub napis „Telefax". Obrazek taki można przygotować za pomocą dowolnego programu graficznego. Tekst przesyłki może być przygotowany za pomocą dołączanego edytora Micro Emacs, sprzedawanego wraz z Amigą na firmowej dyskietce Extras. Jest tu jednak pewien problem z jego wywołaniem — po prostu nie ma jego ikony!!!

Za jedną z ciekawszych i przydatnych funkcji programu MultiFax, uważam możliwość dołączania do nadawanego faksu kolorowego (!) rysunku w formacie IFF. Program rozpoznaje automatycznie grafikę kolorową i odpowiednio „wykropkowywuje" taki rysunek. Dzięki temu można przesyłać np. kolorowe wykresy czy zdigitalizowane obrazy z kamery wideo.

Pewnym niedociągnięciem jest brak pełnej inicjalizacji modemu — w moim przypadku zupełnie niechcący zapisałem w pamięci modemu pewną konfigurację, która uniemożliwiała nadawanie i odbiór faksów (dla wtajemniczonych — zapisałem konfigurację E0 V0, a program potrzebuje E1 V1). Trzeba na to zwracać szczególną uwagę.

Jeżeli można mówić o jakichkolwiek brakach, to zdecydowanie przydałby się w zestawie odpowiedni program instalacyjny pozwalający na przeniesienie całego oprogramowania z dyskietek na dysk twardy. Operacja ta jest dość skomplikowana i początkujący użytkownik może sobie z tym nie poradzić.

Za około 600 marek dostajesz pełnosprawny modem pozwalający Ci także na wysyłanie i odbiór telefaksów. Opcja ta może być bardzo przydatna właścicielom małych firm, dla których zakup normalnego telefaksu jest zbyt kosztowny. Choć sam modem ma szybkość maksymalną 2400 bodów i nie jest wyposażony w MNP, to jednak (moim zdaniem) możliwość pracy jako telefaks rekompensuje częściowo ten brak. Dużą zaletą jest fakt, że urządzenie to można stosować z różnymi typami komputerów przez co koło użytkowników nie jest ograniczone

Użytkownik powinien postarać się o odpowiednie dla danego typu komputera oprogramowanie. Posiadacze Amig mogą wykorzystać całkiem przyzwoite programy dołączane wraz z modemem. Niestety, korzystanie z telefaksu niesie też za sobą pewne wydatki — jeśli chcesz przenieść dokument na papier potrzebna Ci będzie drukarka, praca bez rozszerzonej pamięci RAM jest niemalże niemożliwa, a dysk twardy znacznie przyspiesza operacje. Zestaw wydaje się być atrakcyjny głównie dla małych firm wykorzystujących Amigę do pracy biurowej — cała inwestycja jest relatywnie tania w porównaniu z „dużym" telefaksem.

ZALETY

* Łatwa obsługa programu
* Opcja zamiany rysunku kolorowego na czarno-biały
* Możliwość zaprogramowania godziny nadania faksu
* Możliwość przeglądu dokumentu przed transmisją
* Stosunkowo niska cena
* Możliwość współpracy z dowolnym komputerem korzystającym ze standardowego złącza RS-232
* Dwa urządzenia w jednym (mniejsza liczba kabli na stole)

WADY

* Brak PEŁNEJ inicjalizacji modemu
* Urządzenie nie pracuje z Amigą z prędkością 9600 bodów (ani modem ani telefaks)
* Brak protokołów MNP
* Oprogramowanie zawiera błąd
* Brak programu instalującego konieczne oprogramowanie na dysku twardym.

RAFał Wiosna

Dystrybutorem opisywanego urządzenia jest firma Nogasta Import & Export GmbH, Am Pfeilshof 35e, 2000 Hamburg 65.

OCENA

— Łatwość obsługi	— Średnia (dla początkujących)
— Szybkość oprogramowania	— Wolna (z dyskietki)
— Zgodność ze standardem Hayes (modem)	— Pełna
— Jakość wysyłanych faksów	— Bardzo dobra
— Jakość odbieranych faksów	— Dobra (przeniesienie na papier wymaga wykorzystania drukarki)
— Współpraca z komputerami wyposażonymi w gniazdo RS 232	— Dobra (faks jak i modem)
— Szybkość transmisji	— Zadawalająca (modem max. 2400, faks max 4800 w standardowej konfiguracji)

DANE TECHNICZNE:

MODEM	FAX
0–300 bodów Bell 103	Wysłanie/odbiór do
0–300 bodów CCITT V.21	dowolnego urządzenia
1200 bodów Bell 212A	zgodnego z CCITT
1200 bodów CCITT V.22	Group III Machines
2400 bodów CCITT V.22	9600 CCITT V 29
	(lub 7200, 4800, 2400)

2 gniazda RJ11C zgodne z gniazdami RJ11, RJ12, RJ13 i RJ14.



AMIGA 500

W skład zestawu wchodzą: komputer, zasilacz, myszka, trzy dyskietki 3.5" zawierające: system operacyjny (Workbench 1.3.2), program demonstracyjny „The Very First" oraz programy użytkowe (Amiga Extras). Komplet zawiera także dwie książki — podręcznik wprowadzający oraz opis języka AmigaBasic. Tu można zauważyć pierwszą różnicę między wersją angielską i niemiecką — w wersji niemieckiej kupujący dostaje tylko dwie dyskietki („Workbench" oraz „Extras"), ale za to trzy książki — trzecia wprowadza Czytelnika w tajniki systemu AmigaDOS.

Amiga jest jedynym popularnym w Polsce półprofesjonalnym komputerem 16-bitowym wykorzystującym architekturę wieloprocesową. Dzięki wspomaganiu głównego procesora Motorola 68000 układami specjalizowanymi (grafika, dźwięk, obsługa we/wy) uzyskano bardzo dobre wyniki, niespotykane w komputerach tej klasy cenowej. A na początku Amiga miała być konsolą do gier...

ZASILANIE

Aby zmniejszyć gabaryty komputera i zabezpieczyć sprzęt przed przegrzewaniem zastosowano zasilacz zewnętrzny w którym jednocześnie umieszczono wyłącznik sieciowy. Przewody łączące komputer z zasilaczem są wystarczająco długie aby można było postawić Amigę nawet na dość wysokim stole. Zasilacz generuje trzy napięcia: 5V/2.5A, 12V/1.0A i -12V/0.1A. Zasilacz nie przegrzewa się nawet podczas dość długiej pracy; jego uszkodzenie natomiast można spowodować poprzez dołączenie zbyt dużej liczby dodatkowych urządzeń peryferyjnych (np. kilka napędów).

Warto wspomnieć, że Amiga była sprzedawana kiedyś z tzw. „lekkim" (wagowo) zasilaczem. Był on również wydajniejszy od stosowanego obecnie („ciężkiego"); jeśli więc w Twoim zestawie znajduje się ten drugi typ, staraj się go zbytnio nie przeciążać.

GRAFIKA

W trybie tekstowym Amiga umożliwia uzyskanie 25 linii po 60 lub 80 znaków w wierszu. Ekran jest pozbawiony „ramki" — cechy charakterystycznej dla większości komputerów ośmiobitowych. Możliwa jest swobodna regulacja rozdzielczości pionowej i poziomej obrazu. Standardowa rozdzielczość wynosi 320 x 256 punktów w systemie TV PAL (Europa) lub 320 x 200 dla maszyn pracujących w systemie NTSC (USA, Kanada). Gdy chcemy uzyskać 80 znaków w linii rozdzielczość wzrasta odpowiednio do 640 x 256 (640 x 200) punktów. Grafikę można bez problemu tworzyć i obsługiwać za pośrednictwem instrukcji języka BASIC, który jednak do najszybszych nie należy.

Amiga oferuje paletę 4096 kolorów i specjalny tryb pracy w którym wszystkie te kolory można jednocześnie uzyskać na ekranie. W przypadku większych rozdzielczości pionowych może występować efekt tzw. międzyliniowości (interlace), którego objawem jest drganie obrazu, co NIE jest objawem uszkodzenia komputera.

DŹWIĘK

Amiga jest wyposażona w cztery niezależne przetworniki cyfrowo-analogowe pracujące w wydaniu stereofonicznym, a po polsku mówiąc jest to uproszczona wersja dysku kompaktowego. Język BASIC zawiera funkcję TRANSFORM() dzięki której Amiga może „wymawiać" słowa czy zdania (jest to jak gdyby instalacja starego poczciwego programu SAM znanego z C-64).

KLAWIATURA

Jest wykonana w układzie QWERTY i obsługiwana za pomocą oddzielnego mikroprocesora. W wersji angielskiej kłopoty ze znajdowaniem znaków nie występują czego nie można powiedzieć o wersji np. niemieckiej wyposażonej w klawisze ze znakami niemieckimi. Problem zaczyna się w momencie w którym uruchamiasz program napisany w Anglii na niemieckiej wersji Amigi; znaki nie zawsze pasują i czasami ciężko je znaleźć (zwłaszcza początkującym). Aby jednak nie ograniczać możliwości komputera użytkownik ma możliwość wyboru (w sposób programowy) praktycznie każdego zestawu znaków dla swoich potrzeb.

Sama klawiatura została zbudowana w oparciu o gumę przewodzącą; choć nie jest to konstrukcja najpewniejsza, to jednak żywotność klawiatury jest dość długa.

Ponieważ zakładano, iż Amiga będzie dążyć w kierunku IBM klawiatura została wyposażona w oddzielny blok numeryczny spełniający wymogi PC. Stąd też tajemnicze symbole typu NumL, Home, Pg Dn i podobne — mają one swoje zastosowanie właśnie wtedy, gdy komputer jest wykorzystywany jako klon IBM.

URZĄDZENIA ZEWNĘTRZNE

Amiga może współpracować z monitorem monochromatycznym (np. polski monitor Neptun 156), telewizorem kolorowym bądź czarno-białym wyposażonym w dekoder systemu PAL oraz z monitorem kolorowym w standardzie RGB. Możliwe jest także przyłączanie Amigi do telewizorów wyposażonych w Eurozłącze i uzyskiwanie na nich jakości obrazu odpowiadającej monitorom kolorowym (Eurozłącze zawiera linie sygnałowe RGB). Jakość obrazu uzyskiwanego na telewizorze kolorowym zależy w dużej mierze od jakości telewizora i modulatora; w porównaniu z monitorem obraz OTV kolory są zwykle nieco rozlane.

Amiga 500 jest wyposażona w dwustronny napęd 3.5" zainstalowany w obudowie komputera. Po sformatowaniu dyskietka ma pojemność 850 KB (880 KB bez ścieżek systemowych). Duża gęstość zapisu (80 ścieżek po 11 sektorów) i stosunkowo mała powierzchnia dyskietki zmusza do stosowania raczej dyskietek firmowych zwłaszcza do zapisu ważnych danych.

Formatowanie dyskietki z poziomu Workbench ma pewne wady. Chodzi mi głównie o to, że operacja formatowania jest przerywana w chwili napotkania uszkodzonego fragmentu nośnika zamiast zablokowania danej ścieżki jak ma to miejsce np. w MS-DOS. Do niewątpliwych zalet należy natomiast fakt, że kontroler stacji jest w stanie (przy odpowiednim oprogramowaniu) odczytywać formaty innych dyskietek w tym Atari ST, IBM oraz Apple. Dzięki temu możliwe jest łatwe przenoszenie danych z innych komputerów do Amigi i odwrotnie.

WYJŚCIA

Amiga 500 jest wyposażona w następujące porty:

* dla monitora monochromatycznego
* dla modulatora TV lub monitora kolorowego (standard RGB)
* port równoległy (Centronics)
* port szeregowy (RS-232)
* złącze do przyłączania dodatkowej stacji dysków
* dwa wyjścia fonii (lewy i prawy kanał)
* dwa porty do przyłączenia myszki i/lub joysticka
* gniazdo zasilania
* złącze systemowe ukryte pod klapką z lewej strony
* złącze do rozszerzania RAM zlokalizowane pod pokrywą w dolnej części obudowy komputera



Zdecydowaną zaletą jest fakt, iż wszystkie niezbędne interfejsy są zaintstalowane bezpośrednio na płycie. Tym niemniej nie ma róży bez kolców i to porządnych: port równoległy i port szeregowy zostały przez producenta odpowiednio ZMODYFIKOWANE co uniemożliwia praktycznie bezpieczne korzystnie z tych portów przy zastosowaniu STANDARDOWYCH przewodów. W porcie równoległym modyfikacja ta odbyła się na nóżce 14 (dołączone napięcia +5V), w porcie szeregowym na nóżkach 9 i 10 (napięcia +12V, -12V). Choć instrukcja obsługi ostrzega, że napięcia te są przeznaczone dla sprzętu, który tego POTRZEBUJE, to jednak jest to moim zdaniem ostrzeżenie niedostateczne. W praktyce oba porty wymagają stosowania przewodów robionych na zamówienie lub dostarczonych z urządzeniem peryferyjnym.

PAMIĘĆ RAM

Amiga 500 jest standardowo wyposażona w 512 KB RAM co dla tej klasy komputera nie jest ilością wystarczającą. W wielu zastosowaniach graficznych i muzycznych pamięci tej trzeba znacznie więcej co zmusza użytkownika do szukania odpowiedniego rozszerzenia. Instalację dodatkowej pamięci RAM przeprowadza się prosto — w dolnej płycie Amigi jest wnęka, do której (po otworzeniu klapki) należy włożyć specjalną kartę. W to samo miejsce dołącza się specjalny dysk twardy lub sprzętowy emulator IBM.

OPROGRAMOWANIE SYSTEMOWE

Dostarczane na dyskietkach systemowych oprogramowanie korzysta w olbrzymiej większości z techniki rozwijanych menu (pull-down menu) sterowanych myszką, co ułatwia kontakt użytkownika z komputerem. Duża liczba bardzo różnorodnych programów i opcji z jednej strony przygniata, a z drugiej wywołuje małe zamieszanie w głowie użytkownika z którego jedynym wyjściem jest DOKŁADNE przestudiowanie instrukcji obsługi. Jeśli komuś odpowiada ta kie porównanie, to praca z Amigą przypomina trochę system GEOS znany z C-64 lub MS-WINDOWS znane z komputerów IBM.

Większość poleceń systemowych bierze swój rodowód z systemu MS-DOS, zmienione są jedynie niektóre parametry i składnia instrukcji. Konstruktorzy chcieli rzeczywiście przekazać praktycznie wszystkie możliwości w ręce użytkownika — w efekcie trudno jest się w tym połapać (zwłaszcza początkującym). Do najciemniejszych punktów należy szereg katalogów systemowych o różnych nazwach choć wszystko razem można by wsadzić (moim zdaniem) do jednego worka.

ZALETY

* Bardzo dobra grafika i muzyka
* Bogate oprogramowanie
* Dostępne złącza standardowe (niepotrzebne interfejsy)
* Duża elastyczność systemu
* Możliwość wymiany plików z IBM PC, ATARI ST, APPLE

WADY

* Niewystarczająca pamięć w konfiguracji podstawowej (512KB) do zastosowań profesjonalnych
* Odbiegające od standardu sygnały w portach komunikacyjnych
* Niedopracowane formatowanie dyskietek z poziomu Workbench
* Dość zawiła struktura katalogów systemowych

OCENA

Grafika	bardzo dobra, duże możliwości
Dźwięk	bardzo dobry, duże możliwości
Oprogramowanie	bogate, łatwodostępne (gry)
Instrukcja obsługi	przejrzysta
System operacyjny	skomplikowany w użytkowaniu
Główna wada	zbyt mała pamięć RAM
Peryferia	łatwodostępne, duży wybór
Popularność w kraju	duża i stale rosnąca
Zgodność z innymi komputerami	emulatory Atari ST, C-64, IBM, Spectrum, karty dla IBM PC XT/AT
Ocena ogólna	7 w skali 1—10

DANE TECHNICZNE AMIGI

Rozdzielczości obrazu (standardowe)
320 x 256 — 2, 4, 8, 16, 32, 64*, 4096** kolorów
640 x 256 — 2, 4, 8, 16 kolorów
320 x 512 — 2, 4, 8, 16, 32, 64*, 4096** kolorów
640 x 512 — 2, 4, 8, 16 kolorów

Maksymalna rozdzielczość: 700 x 600 przy 16 kolorach
Pamięć CHIP — 512 KB (standard) do 1024KB
FAST — 512 KB do 8 MB
SLOW — 512 KB do ok. 1.8 MB

Częstotliwość zegara — 7. 14 MHz
Dźwięk — 4 niezależne przetworniki cyfrowo-analogowe
Zakres — 8 oktaw

* — tryb Extra Half Brite
** — tryb HAM (Hold And Modify)

Dystrybutorem opisywanego sprzętu jest firma „TAL", Sp. z o.o, Mikowa 45, 02-411 Warszawa Włochy, tel. 662-35-16 (informacje) lub 23-98-53, fax 695-12-35.

RAFał Wiosna

Drodzy Czytelnicy!

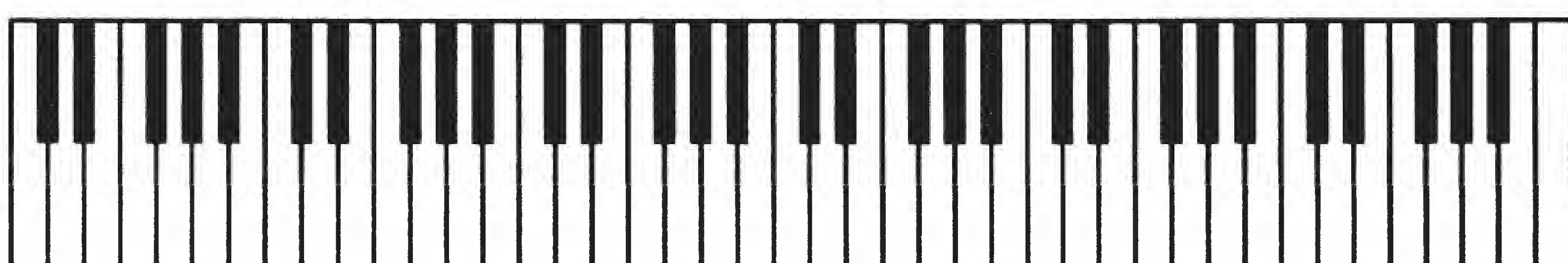
Dostajemy od Was co miesiąc szereg listów z pytaniami czy „KLANU COMMODORE" nie można by w jakiś sposób rozszerzyć. W związku z tym zdecydowano, że wzorem „Mojego Atari" powstanie nowe czasopismo poświęcone wyłącznie komputerom Commodore — „C&A". Liczymy, że zaspokoi ono Wasze potrzeby.

Nowy magazyn ukaże się po raz pierwszy już w październiku tego roku I, w zależności od Waszego zainteresowania, będzie wydawany cyklicznie lub nie. W pierwszym numerze znajdziecie szereg materiałów przeznaczonych dla początkujących; artykuły te będą w rzeczywistości obszernymi odpowiedziami na najczęściej spotykane w Waszych listach pytania.

Jednocześnie zapraszamy wszystkich zainteresowanych do współpracy i współredagowania z nami nowego czasopisma. Wszelkie informacje szczegółowe można uzyskać telefonicznie pod numerami 21-12-05 i 643-1840 w godzinach 10—16.
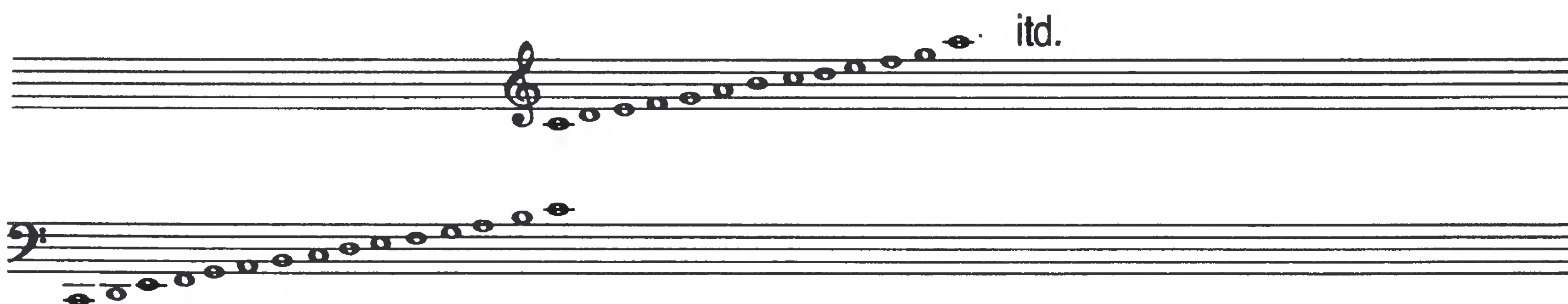
Klaudiusz Dybowski

kolejno czarne klawisze w oktawie: cis, dis, fis, gis, ais lub des, es, ges, as, b



wielka mała razkreślna dwukreślna trzykreślna czterokreślna

C D E F G A H c d e f g a h c_1 . . . a_2 c_5



Muzyka od kuchni

Commodore 64/128 i Amiga to komputery zdecydowanie wyróżniające się spośród innych bardzo dużymi możliwościami dźwiękowymi. Walory te można łatwo wykorzystać pod warunkiem, że użytkownik dysponuje odpowiednim oprogramowaniem i pewnym doświadczeniem z zakresu kompozycji; stąd też „Klan Commodore" postanowił rzecz całą zbadać „od kuchni".

W cyklu tym postaramy się przedstawić nie tylko elementarne podstawy muzyki ale również programy zamieniające Twój komputer w syntetyzer czy elektroniczną perkusję. Wierzymy, że i tym razem Czytelnicy nie zawiodą i wspomogą nas swoimi doświadczeniami i materiałami.

Redakcja

Jak głosi anegdota, pewnemu klasykowi zadano pytanie czy nauka gry na fortepianie jest zadaniem prostym. „Jest to nawet wyjątkowo proste, droga pani" — odparł zagadnięty — „wystarczy tylko w odpowiednim momencie wcisnąć właściwy klawisz, a całą resztę zrobi już sam fortepian..."

Muzyka, jak każda inna dziedzina twórcza rządzi się swymi własnymi prawami i zasadami. Dla ułatwienia możemy przyjąć, że komponowanie to nic innego jak PROGRAMOWANIE dźwięku, co jednocześnie wyjaśnia dlaczego komputer tak dobrze nadaje się zarówno do kompozycji jak i odtwarzania. Zwróć uwagę, że komputer i muzyka mają jedną wspólną cechę — jest nią działanie w czasie. Z elektronicznego punktu widzenia nowoczesny syntetyzer to po prostu specjalizowany komputer; dane o instrumentach jakie może on naśladować są zapisywane w zwykłej pamięci RAM i/lub ROM. Pojedyncze nuty nawet najbardziej skomplikowanego utworu muzycznego możemy przyrównać do poszczególnych instrukcji zawartych w zwykłym programie komputerowym; analogie, jak widać, są uderzające.

Wiele osób sądzi, że nauka podstaw muzyki to coś bardzo trudnego i niepotrzebnego — przecież na fortepianie można grać także bez znajomości nut, „na słuch". Takie podejście kojarzy mi się jednak z układaniem programu komputerowego w jakimkolwiek języku programowania bez dokładnej znajomości składni tego języka. Spróbujmy zatem przekonać się, czy zapis nutowy jest rzeczywiście tak skomplikowany i trudny; dobrze byłoby, gdybyś przedstawione tu przykłady mógł od ręki wpisywać i wypróbowywać na swym komputerze.

Dźwięki i ich nazwy

W muzyce przyjęło się, że cały obszar częstotliwości dostępny dla ludzkiego słuchu (ok. 20 — 20000 Hz) podzielony jest na malutkie cząstki zwane półtonami. Półton to najmniejsza odległość między dźwiękami (różnica wysokości) jaką jest w stanie wychwycić nasze ucho. Dwa tony oddalone od siebie o mniej niż półton słyszymy już jako jeden „nieczysty" dźwięk. Dwanaście półtonów tworzy jedną oktawę. Jak wiadomo, dwa dźwięki w odstępie oktawy brzmią identycznie (stosunek częstotliwości wynosi 1:2), a więc, aby opisać całą słyszalną skalę wystarczy nadać nazwy tylko dwunastu dźwiękom w obrębie oktawy, a poszczególne oktawy oznakować. (rys. 1).

Klawisze białe określa się kolejnymi literami alfabetu, przy czym za punkt wyjściowy przyjęto częstotliwość 440 Hz (klawisz a1). Nazwy oktaw to kolejno od dołu: subkontra, kontra, wielka, mała, raz-, dwu-, trzy-, cztero- i pięciokreślna. Z pewnych względów w niektórych krajach (m.in. w Polsce) literą *b* oznacza się klawisz czarny nad *a*, dlatego też klawisz biały między *a* i *c* oznaczono literą *h*. W krajach anglosaskich (skąd pochodzi większość programów) nasze *H* określa się jako *B*, a nasze *B* — jako *B* flat).

Linie dodane i klucze

Rys. 1 przedstawia część typowej klawiatury fortepianowej oraz zapis nutowy dźwięków odpowiadających białym klawiszom (bez uwzględnienia rytmu). Ponieważ pięciolinia zawiera zbyt mało miejsca, aby zapisać wszystkie możliwe wysokości nut, stosuje się tzw. linie dodane dolne i górne (np. dźwięk c3 leży na drugiej linii dodanej górnej) oraz różne klucze. Klucz basowy umożliwia zapisywanie oktaw niższych, zaś klucz wiolinowy (przed dźwiękiem c1) — oktaw wyższych. Dla orientacji wystarczy zapamiętać, że dźwięk c1 w kluczu basowym leży na pierwszej linii dodanej górnej, a w kluczu wiolinowym na pierwszej linii dodanej dolnej.

Krzyżyki i bemole
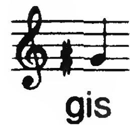
Aby dołączyć do zapisu również czarne klawisze, musimy skorzystać z krzyżyków lub bemoli. Krzyżyk (rys. 2a) podwyższa dany dźwięk o półton, natomiast bemol (rys. 2b) obniża go o tą samą wartość. Nuta z krzyżykiem otrzymuje w nazwie końcówkę -is (np. *c* z krzyżykiem to cis), a z bemolem — końcówkę -es (*d* z bemolem = des). Zarówno krzyżyk jak i bemol mogą stać przy kluczu co oznacza, że obowiązują wówczas na całej długości pięciolinii. Jeśli krzyżyk lub bemol stoi bezpośrednio przy nucie traci on swą ważność wraz z końcem taktu. Oba znaki można zlikwidować kasownikiem (rys. 2c).

Rytm i metrum

Istotą wszelkiej muzyki jest ruch dźwięków, czyli przebieg w czasie — nawet pojedyńczy dźwięk trwa przecież określoną chwilę. Rytm określa długość trwania poszczególnych dźwięków. Najdłużej brzmi cała nuta, dwa razy krócej półnuta, cztery razy kró-
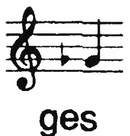
cej ćwierćnuta itd. — patrz rysunek 3.

Rola metrum to organizacja całego przebiegu muzycznego w równe pod względem czasu trwania odcinki zwane taktami.



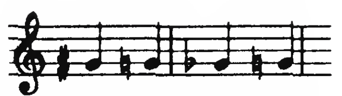
gis

Aby określić metrum podajemy przy kluczu dwie cyfry. Górna cyfra mówi nam ile jednostek rytmicznych określonych cyfrą dolną pomieścić może każdy takt aż do końca utworu. Zatem metrum (rys. 4a) oznacza,



ges

że dowolny takt zawierać będzie równowartość rytmiczną czterech ćwierćnut (a więc np. jedną całą nutę, osiem ósemek itd.). Inne najczęściej spotykane metrum to: 3/4 — trzy ćwierćnuty w takcie (rys. 4b) i 3/8 — trzy ósemki w takcie (rys. 4c). Me-



trum obowiązuje na całej długości zapisu nutowego (czyli w całym utworze).

Kropka przy nucie przedłuża jej wartość o połowę (czyli np. ćwierćnuta z kropką = ćwierćnuta + ósemka), a druga kropka o połowę połowy (np. ósemka z dwoma kropkami = ósemka + szesnastka + trzydziestodwójka — patrz rys. 5).

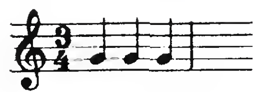
Innym sposobem przedłużenia wartości rytmicznej jest połączenie łukiem dwóch lub więcej nut. Ma to zastosowanie wtedy, gdy czas trwania danego dźwięku przekro-

𝅝 = 𝅗𝅥 + 𝅗𝅥 = ♩ + ♩ + ♩ + ♩ =
= ♪+♪+♪+♪+♪+♪+♪+♪=
= 𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯+𝅘𝅥𝅯

czyłby ramy taktu. Załóżmy, że chcemy zapisać trzy półnuty w takcie (rys. 6). Takie metrum mieści jednak tylko trzy ćwierćnuty, zatem połowę drugiej półnuty (czyli ćwierćnutową „nadwyżkę") musimy przenieść do drugiego taktu.

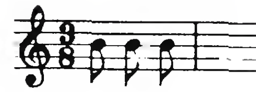


Stosując tylko podziały rytmiczne omawiane dotychczas, bylibyśmy mocno ograniczeni, bowiem podział czasu trwania dźwięku na 1/2, 1/4 czy 1/16 nie pozwala nam zapisać wszystkich możliwych układów rytmicznych. Dlatego też do zapisy-



wania bardziej skomplikowanych rytmów używamy specjalnych grup rytmicznych zwanych odpowiednio: triola (albo trójka — podział na trzy), kwintola (albo piątka — podział na pięć) lub też septymola (siódemka) itd. W muzyce klasycznej spotkać można bardzo często takie nieregularne podziały, np. jedenastki, osiemnastki itp. Jak to wygląda w zapisie ilustruje rysunek 7. Nieregularną grupę oznaczamy na górze liczbą, która mówi nam na ile podzielona została wartość podstawowa, a pierwszą i ostatnią nutę łączymy łukiem (to ostatnie nie jest konieczne).

Kończąc rozważania o rytmie słów kilka o pauzach. Pauza, jak sama nazwa wskazuje, to po prostu przerwa, chwila ciszy w muzyce. Jedyną cechą charakteryzującą pauzy jest czas ich trwania czyli wartość rytmiczna. Tak więc pauzy mogą być całonutowe, półnutowe, ćwierćnutowe, ósemkowe, trójkowe, siódemkowe itd. Co więcej, każda pauza może być przedłużona za



pomocą jednej lub dwóch kropek na tych samych zasadach co „zwykłe" nuty.

Tempo

Aby w pełni określić przebieg w czasie poszczególnych dźwięków należy jeszcze dokładnie oznaczyć tempo. Musimy sobie bowiem uświadomić, że ten sam rytm wykonany w różnych tempach będzie miał różne czasy trwania. W większości programów dla C-64 tempo ustala sam komputer; programy bardziej ambitne pozwalają na określanie tempa za pomocą nazw trady-



cyjnie używanych w muzyce, a zapożyczonych z języka włoskiego. Oto niektóre z nich: Largo, Adagio, Lento (tempa wolne), Moderato, Andantino (średnioszybkie), Allegro, Vivo, Presto (szybkie i bardzo szybkie).

Notacja wielogłosów

Do zapisu utworów trzygłosowych (równoczesne prowadzenie trzech głosów umożliwia nam wspaniale zaprojektowany generator dźwięków w C-64) stosujemy



dwie pięciolinie powiązane w jeden system, przy czym w większości przypadków na pięciolinii górnej obowiązuje klucz wiolinowy, a na dolnej basowy. Dźwięki leżące w linii pionowej grane są równocześnie. Jeśli na jednej pięciolinii występują dwa głosy, to nuty głosu górnego powinny mieć laseczki skierowane do góry, a głosu dolnego — do dołu.



Nieco inaczej wygląda sprawa z zapisem dźwięków o nieustalonej wysokości — najlepszym przykładem jest tu oczywiście perkusja a raczej to, co większość ludzi rozumie potocznie pod tym pojęciem. Do tego tematu powrócę przy omawianiu poszczególnych programów muzycznych i stosowanych w nich sposobów zapisu dźwięku.

Christian Grzenkowicz

Dokończenie ze str. 20

WYCISKANIE WODY

nalnego obrazka. Czy to dużo, czy mało? Raczej niewiele. Wszystkie programy archiwizujące (pkarc, pkzip, lha etc.) dają nieco lepsze wyniki, w dodatku efekt ich działania nie jest tak mocno zależny od stopnia „zaczernienia". W kilku przypadkach jednak zaprojektowany algorytm był bezkonkurencyjny, zwłaszcza wtedy, gdy poddawane kompresji obrazki były niemal całkiem białe.

Istnieją dwa sposoby na to, by prezentowany algorytm stał się nieco wydajniejszy. Pierwszy jest stosunkowo prosty — wystarczy zauważyć, że jeśli w trzech pierwszych ćwiartkach badanego aktualnie kawałka nie było żadnych czarnych punktów, muszą one być w czwartym i nie trzeba już wysyłać bitu równego jeden jako sygnału — ta ćwiartka nie jest pusta. W ten sposób można zaoszczędzić kilka procent.

Drugi sposób ma pewną wadę — jego zastosowanie nie daje gwarancji, że oryginalny obrazek uda się odtworzyć ze stuprocentową dokładnością. W przypadku danych innych niż graficzne, niejednoznaczna metoda kompresji i dekompresji oznacza katastrofę. W przypadku rysunku można sobie jednak pozwolić na pewne drobne zmiany — zwłaszcza, że oko ma zdolności do wygładzania koślawych krzywych i rozpoznawania kształtów nawet wtedy, gdy uległy niewielkiej deformacji. Można więc spróbować zastąpić funkcję boxempty (pudełko puste) funkcją boxalmostempty (pudełko prawie puste), która będzie zwracać wartość TRUE gdy badany przez nią obszar będzie zawierać mniej niż np. 1% czarnych punktów. Może to dać olbrzymi zysk podczas kompresji, choć ponad wszelką wątpliwość spowolni ją kilkakrotnie. W obecnej wersji bowiem sprawdzanie stanu obszaru — pusty czy nie — wystarczy prowadzić do momentu napotkania pierwszego czarnego punktu. W przypadku funkcji boxalmostempty należałoby każdorazowo sprawdzać stan większej liczby punktów, co spowolni jej działanie.

Wymyślony algorytm nie jest więc rewelacyjny i nie spowoduje przewrotu w dziedzinie kompresji danych. Powiem więcej — nie jest taki całkiem nowy, można go bowiem uznać za wariant tzw. algorytmów autoadaptacyjnych (tym samym okaże się, że po raz kolejny udało mi się wyważyć otwarte drzwi). Myślę jednak, że elegancja (oparta na rekurencji) uzasadnia przedstawienie go szerszej publiczności. Jeżeli ktoś spróbuje usprawnić algorytm (zgodnie z podanymi powyżej pomysłami) bardzo chętnie o tym napiszę. W końcu programowanie (i wymyślanie nowych algorytmów) może być znakomitą formą relaksu, do czego namawia

Marcin Borkowski

Na wydruku znajdują się tylko najważniejsze — z punktu widzenia algorytmu — procedury. Nie ma dwóch procedur bardzo istotnych z punktu widzenia systemu — send i get, które powinny wysyłać (lub pobierać) do pliku zewnętrznego, zawierającego poddany kompresji obraz, binarne jedynki i zera.

```
function boxempty(x1,y1,xg,yg : integer): boolean;
var
  x,y : integer;
begin
  boxempty:=false;
  for x:=x1 to xg do
    for y:=y1 to yg do
      if GetPoint(x,y)=1 then EXIT;
  boxempty:=true
end;

procedure scrcompres(x1,y1,xg,yg : integer);
var
  dx2,dy2  :  integer;
begin
  if boxempty(x1,y1,xg,yg) then send(0)
  else
  begin
    send(1);
    if (xg<>x1) or (yg<>y1) then
    begin
      dx2:=(xg-x1) div 2;
      dy2:=(yg-y1) div 2;
      scrcompres(x1,y1,x1+dx2,y1+dy2);
      if xg<>x1 then scrcompres(x1+dx2+1,y1,xg,y1+dy2);
      if yg<>y1 then scrcompres(x1,y1+dy2+1,x1+dx2,yg);
      if (xg<>x1) and (yg<>y1) then
                    scrcompres(x1+dx2+1,y1+dy2+1,xg,yg)
    end
  end
end;

procedure scrdecomp(x1,y1,xg,yg : integer);
var
  dx2,dy2  :  integer;
  bit      :  byte;
begin
  get(bit);
  if bit=0 then EXIT;
  if (xg=x1) and (yg=y1) then PutPixel(x1,y1,1)
  else
  begin
    dx2:=(xg-x1) div 2;
    dy2:=(yg-y1) div 2;
    scrdecomp(x1,y1,x1+dx2,y1+dy2);
    if xg<>x1 then scrdecomp(x1+dx2+1,y1,xg,y1+dy2);
    if yg<>y1 then scrdecomp(x1,y1+dy2+1,x1+dx2,yg);
    if (xg<>x1) and (y1<>yg) then
                  scrdecomp(x1+dx2+1,y1+dy2+1,xg,yg)
  end
end;
```

BAZY i nie tylko

W numerze poświęconym bazom danych i obsługującym je programom, klan IBM-a mógłby bez żadnych problemów zająć cały numer, plus kilkanaście stron dodatku nadzwyczajnego. Nie ma w tym nic dziwnego — PC-ety są standardem komputerów biurowych, i jako takie służą w większości przypadków właśnie do obsługi baz danych. Natura nie znosi próżni, więc tam, gdzie istnieje popyt, pojawia się duża podaż różnorakich produktów. Liczących się standardów baz danych na PC jest cztery albo pięć, programów, które warto opisać — nawet kilkanaście. Nie sposób tego wszystkiego zmieścić na czterech kolumnach, które mamy do dyspozycji, zwłaszcza, że same bazy to nie wszystko — istnieje wiele programów wspomagających, użytecznych przy różnych operacjach. Dla kogoś nie zajmującego się profesjonalnie opracowywaniem baz danych korzystanie z nich (albo nie) zależy od prywatnych upodobań.

Dlatego też wybór opisywanych programów jest niezwkle subiektywny i wynika raczej z naszych przyzwyczajeń niż z próby pełniejszego pokazania istniejącego rynku. Z tego względu zabraknie wśród naszych artykułów tekstu na temat programu Nutshell, skądinąd bardzo sympatycznego, nie będzie też mowy o Paradox-ie. Nie oznacza to, że są to programy gorsze od opisywanych — są po prostu inne. Jeżeli uważacie, że program, którego używacie, powinien zostać przedstawiony — napiszcie o nim do nas, do tematu baz danych chętnie kiedyś wrócimy.

Mimo że najpopularniejszym na świecie standardem jest bez wątpienia dBase, nie znajdziecie go w klanie, ale tylko dlatego, że został już opisany w innej części numeru. DBase, choć znakomity do wielu zastosowań, ma jednak kilka wad, w tym najważniejszą z naszego punktu widzenia jest brak polskich liter. Korzystanie z nich wymaga gimnastyki i nie zawsze jest możliwe (np. podczas sortowania). Podobnie sprawa ma się z Clipperem. Rozwiązaniem może być program TIG — prosta, kartotekowa baza danych, spokrewniona z TAG-iem. TIG ma wprawdzie własny format zbioru danych, jednak pozwala na eksport i import danych w formacie dBase. Możliwości TIG-a są znacznie skromniejsze niż dBase, ale w wielu przypadkach powinny być absolutnie wystarczające, przy dodatkowym plusie w postaci polskiego interfejsu użytkownika. W chwili, w której piszę te słowa (koniec maja), nie wiem jeszcze, kiedy TIG pojawi się w sprzedaży — według zapowiedzi firmy InfoService ma to nastąpić przed końcem tego kwartału.

W grudniu ubiegłego roku opisywałem program MkS_vir. Od tamtego czasu upłynęło sporo wody, przybyło trochę wirusów, a sam program zmienił właściciela — jest nim obecnie warszawski Apexim (tel. 38-56-13). Najnowsza posiadana przez nas wersja MkS_vir-a (3.22) prezentuje się znacznie lepiej niż zeszłoroczne — oprócz koniecznych ze względu na nowe wirusy zmian w samym programie (i nowego wyglądu wizytówki programu) przybyła mu elegancka instrukcja i solidna kartonowa koperta.

Wszystkich producentów oryginalnego oprogramowania zapraszam do informowania nas o posiadanych programach (kopie demonstracyjne lub do testów mile widziane, osbiście gwarantuję, że nie opuszczą redakcji) — chętnie o nich napiszemy przynajmniej kilka słów.

Marcin Borkowski

TURBO DATA BASE

Z Pracowni Programistycznej Pana Jerzego Wawro otrzymaliśmy pakiet Turbo Data Base. Wszystko zaczęło się od naszych prenumeratorów — żeby móc ich obsłużyć, potrzebowaliśmy odpowiedniego programu. Tak się składa, że mamy kilku dobrych programistów piszących w Pascalu, zależało nam jednak na wygodnym dostępie do naszych danych poza programem — stąd wziął się pomysł, żeby skorzystać z formatu dBase. Takie rozwiązanie wymaga odpowiedniego narzędzia, pozwalającego na łatwe pisanie w Turbo Pascalu programów obsługujących zbiory dBase — tu na horyzoncie pojawia się Pan Wawro, oferujący taki właśnie zestaw narzędzi.

Przyznam, że rozmiary pakietu stanowiły dla mnie duże zaskoczenie. Spodziewałem się bowiem, że otrzymamy zestaw kilkunastu, może kilkudziesięciu procedur, pozwalających na dostęp do zbiorów zapisanych przez dBase i automatyczne rozpoznawanie struktury rekordów. Tymczasem TDB składa się z kilkunastu modułów (skompilowanych przy użyciu Turbo Pascala 5.5), stanowiących szerokie środowisko pracy, pozwalające nie tylko na korzystanie z plików dBase, ale także na tworzenie we własnych programach specyficznego interfejsu użytkownika.

Pakiet zajmuje cztery dyskietki, na których oprócz samych modułów znajduje się zarchiwizowana dokumentacja, przykładowy program, kilka programów pomocniczych i TB, czyli uniwersalny program 'shareware' służący do obsługi dowolnych baz danych w standardzie dBase. Dokładne poznanie całego pakietu może zająć kilka tygodni, toteż postaram się zwrócić uwagę wyłącznie na te jego elementy, które uznałem za najciekawsze — i nie ukrywam, że wybór jest subiektywny.

Najistotniejszą częścią TDB są biblioteki do obsługi plików *.dbf. Jest ich kilka o różnym stopniu zaawansowania — pozwala to na korzystanie z mniej lub bardziej wyrafinowanych metod dostępu do danych zawartych w jednym lub kilku plikach. Na próbę postanowiłem korzystając z omawianych modułów napisać program wypisujący na ekranie zawartość dowolnego pliku dBase, podawanego jako parametr wywołania. Program miał wypisywać na ekranie nazwy pól i ich zawartość dla wszystkich rekordów. Innych przydatnych opcji (przeszukiwanie bazy, sortowanie lub indeksowanie — wszystko dostępne jako procedury) nie próbowałem w tym momencie wykorzystywać, by nie komplikować zanadto zagadnienia. Napisanie programu zajęło mi około godziny — wystarczyło 35 linii, by zadanie zostało zrealizowane.

Program widoczny na wydruku bynajmniej nie jest reprezentatywny dla tego, co można zrobić za pomocą TDB. Pokazuje jednak najważniejszą cechę pakietu — ktoś znający dobrze Turbo Pascala może w nim napisać części programu odpowiedzialne za komunikację z użytkownikiem, może w TP zaimplementować wszystkie operacje wejścia-wyjścia, nie przejmując się koniecznością poznania formatu w jakim dBase zapisuje dane, ani koniecznością implementowania algorytmów indeksujących bazę itd. Oczywiście taki sam efekt końcowy można uzyskać korzystając z samego dBase lub Clipper-a, jednak wymaga to bliższego poznania tych programów, co nie zawsze musi być opłacalną inwestycją.

Drugą częścią pakietu jest zbiór narzędzi służących do projektowania i generowania konkretnych aplikacji. W jego skład wchodzi kilka programów, pozwalających na zaprojektowanie i analizę bazy danych (kartotekowej lub relacyjnej), wielopoziomowego menu i systemu podpowiedzi, a także wydruków. Przygotowane za pomocą tych narzędzi dane zostają zapisane na dysku w postaci plików, których format jest znany innym elementom systemu. W zamyśle ma to ułatwić prace projektowe nad nowymi systemami.

Jak dobrym narzędziem jest cały pakiet? Trudno udzielić jednoznacznej opowiedzi na takie pytanie. W zależności od potrzeb będzie on albo zbyt rozbudowany, albo zbyt skromny. Dla kogoś potrzebującego wyłącznie dostępu do plików dBase, a w takiej sytuacji jest większość programistów dysponujących dobrze sobie znanymi zestawami narzędzi, co najmniej połowa pakietu jest zbędna. Dla kogoś potrzebującego dużego pakietu do generacji aplikacji, proponowane narzędzia wydają się zbyt skromne i za mało wygodne, w porównaniu choćby z biblioteką Turbo Professional.

Dodatkowym mankamentem TDB jest mała przyjazność wchodzących w jego skład programów i dokumentacji. Podczas pisania zamieszczonego obok programu musiałem metodą prób i błędów sprawdzać, w jaki sposób powinienem skorzystać z funkcji cField, iField i lField, zwracających zawartość poszczególnych pól rekordu. Nigdzie w opisie nie znalazłem bowiem komentarza tłumaczącego, jak należy to zrobić. Nawet program przykładowy (bg.pas), który mógłby posłużyć do rozwiania moich wątpliwości, nie dał się wykorzystać — na dyskietkach zabrakło bowiem plików danych, które mógłbym przy jego pomocy obsłużyć. Wprawdzie jaka taka znajomość dBase pomogła mi w opanowaniu sytuacji, ale dalej nie wiem dlaczego napisany przeze mnie program działa prawidłowo (i czy nie padnie w jakiejś specyficznej sytuacji). W zamieszczonych w opisie przykładach zdarzają się odwołania do nieistniejących w pakiecie procedur, co więcej, zdarza się to również w dokumentacji technicznej.

Już podczas pisania tego tekstu chciałem komuś pokazać przydatność TDB do tworzenia własnych programów korzystających z plików dBase i uruchomiłem program którego wydruk zamieściłem obok. Jako parametr wejściowy podałem nazwę jakiegoś pliku, którego nie wykorzystywałem wcześniej w tym celu, no i stało się — zgodnie z prawami Murphy'ego program przetestowany na kilku zestawach danych padł na pierwszym innym zestawie danych podczas oficjalnej prezentacji swego działania. Przeprowadzone śledztwo wykazało, że przyczyną tego faktu było pole typu MEMO. Wprawdzie w dokumentacji TDB jest napisane, że pola MEMO nie są przez pakiet obsługiwane, ale nigdzie nie było informacji, że próba otwarcia pliku zawierającego takie pola zakończy się błędem opisanym w dokumentacji jako: Plik nie jest typu 'DBF'!

Prace nad naszym programem obsługi prenumeraty jeszcze trwają, więc nie chcę na razie wydawać ostatecznego werdyktu o TDB. Nie wiem jeszcze jak szybki jest pakiet, i jakie jeszcze niespodzianki spotkają nas podczas prób jego opanowania. Wiem jednak już na pewno, że korzystanie z niego nie jest tak proste, jak mogłoby być.

Marcin Borkowski

```
uses db0,crt;

var
  i          : integer;
  c          : char;
  s          : string;

begin
  s:=paramstr(1)+'.';  s[0]:=chr(pos('.',s)-1);
  dbfopen(1,s);                   { otwórz plik podany jako parametr }
  with area[1] do
  begin
    repeat
      clrscr;
      writeln('Plik: ',s,'liczba rekordów: ':32,OpenNum:5,
              ' Rekord: ',CurrentRecNo:5,#13#10);
      for i:=1 to HD.NumFields do with HD.fields^[i] do
      begin
        write(Fname:20,' ');                              { nazwa pola }
        case Ftype of
          'C'  :  writeln(sField(1,i));
          'N'  :  if DPlength=0 then writeln(lField(1,i))
                  else writeln(rField(1,i):0:DPlength);
          'L'  :  writeln(blField(1,i));      { brak pola typu DATA }
        end;
      end;
      c:=readkey;
      if c=#0 then c:=readkey;
      if c=#27 then HALT;
      nextrecord(1);                       { wozytaj kolejny rekord }
    until dEof                                { koniec pliku *.dbf }
  end;
  dbfClose(1);                                       { zamknij plik }
  clrscr
end.
```

Producentem Turbo Data Base jest:
Jerzy Wawro
Pracownia Programistyczna
Os. Niepodległości 2/16
37-500 Jarosław



dBASE jest klasycznym interpreterem, tak samo jak większość implementacji BASIC-a. Każde polecenie użytkownika musi być przed wykonaniem rozpoznane i zanalizowane. Dotyczy to zarówno pojedynczych komend jak i programów napisanych w specyficznym języku, składającym się z rozkazów dBASE. Taki system jest zwykle wolny, stąd pojawiło się zapotrzebowanie na kompilator. Rolę tę spełnia rozwijany równolegle z systemem dBASE program Clipper firmy Nantucket. Jego kolejne wersje odpowiadały kolejnym wersjom systemu dBASE: Clipper Summer '85 — dBASE III, Clipper Summer '87 — dBASE III Plus. Aktualnie jest rozpowszechniana najnowsza wersja — Clipper 5.0.

Najbardziej znana i rozpowszechniona w Polsce jest wersja Summer '87. Pozwala ona (z niewielkimi zastrzeżeniami) kompilować programy napisane w języku dBASE III plus. Przede wszystkim jednak można Clippera traktować jako narzędzie niezależne od dBASE, wówczas użytkownik dostaje do swojej dyspozycji całą gamę rozszerzeń, nowych komend i funkcji. Najważniejsze z nich jak również pewne niewielkie ograniczenia (w stosunku do dBASE III plus) pokrótce omówię.

Zacznijmy od ograniczeń. Po pierwsze, brak realizacji w Clipperze tzw. komend edycji pełnoekranowych (EDIT, INSERT, CHANGE, APPEND), programista musi sam zaprojektować i realizować edycję i wprowadzanie danych. Po drugie, inaczej działają polecenia LIST i DISPLAY — w Clipperze należy wyspecyfikować listę pól, które mają być wyświetlane bądź drukowane (w dBASE domyślnie są to wszystkie pola). Oznacza to, że bez naniesienia poprawek nie da się za pomocą Clippera skompilować żadnego programu, zawierającego któreś z wymienionych poleceń.

W porównaniu z listą rozszerzeń, powyższe ograniczenie przestaje mieć wielkie znaczenie:

— można otworzyć jednocześnie do 255 plików (DOS 3.30), każdy w innym obszarze roboczym (w dBASE III plus — 15),

— liczba aktywnych indeksów w danym obszarze roboczym — do 15 (w dBASE III plus najwyżej 7),

— zwiększenie maksymalnej długości pól znakowych, pól w pliku notatnikowym (MEMO), zniesienie ograniczeń długości rekordu i liczby pól w rekordzie:

	dBASE III Plus	Clipper '87
Pole typu znakowego (CHARACTER)	254	32 K
Memo	5000	64 K
długość rekordu	4000	ograniczone jedynie przez dostępną pamięć komputera
l. pól w rekordzie	128	

Kolejne rozszerzenia dotyczą języka programowania Clipper. Najważniejsze z nich to:

— możliwość definiowania własnych funkcji (w dBase tylko procedur),

— możliwość tworzenia menu (wybieranie przez podświetlanie właściwej pozycji) — bardzo ważne przy pisaniu programów przyjaznych dla użytkownika,

— możliwość definiowania tablic oraz korzystania z szeregu funkcji związanych z operacjami na tablicach (niestety mogą to być wyłącznie tablice jednowymiarowe),

— możliwość budowania pętli o zadanej liczbie powtórzeń, czyli pętli FOR ... TO ... NEXT; w dBASE jedyny dostępny rodzaj pętli to DO WHILE ‹warunek› ... ENDDO,

— procedury i funkcje widoczne są z każdego poziomu, w każdym z plików procedur (**.prg**), o ile będą włączone w procesie łączenia (konsolidacji); w dBASE III plus z procedur zdefiniowanych w innym pliku może korzystać wyłącznie program główny.

— Clipper umożliwia tworzenie własnych podpowiedzi (HELP) do programu, uaktywnianych klawiszem F1,

— Clipper rozszerza obsługę wprowadzania danych przez komendę GET o testowanie funkcją VALID. W ramach tej funkcji (musi ona zwracać wartość logiczną) można wykonywać skomplikowaną kontrolę wprowadzanej danej, a także dowolne inne operacje, w tym np. udostępniać listę dopuszczalnych wartości i wybierać daną z przedstawionej listy (np. nazwę miesiąca lub dnia tygodnia),

— Clipper '87 dostarcza cały szereg nowych funkcji: związanych z obsługą pól typu MEMO, czytaniem i pisaniem do plików obcych (poza typem właściwym dla dBASE i Clippera — **.dbf**). Umożliwia to współpracę z dowolnymi plikami znakowymi czy nawet binarnymi, związanymi ze środowiskiem działania programu (np. obsługą ekranu — zapamiętywanie i odtwarzanie ekranu lub jego części, obsługą drukarki — z możliwością testowania jej stanu). Znacznie także ułatwione jest korzystanie z DOS-u i sieci.

Dodatkową korzyścią korzystania z kompilatora jest fakt, że kompilacja programu eliminuje wszystkie błędy formalne (składniowe), czego interpreter nie może zapewnić — zawsze bowiem może się zdarzyć, że podczas testowania któraś ze ścieżek którymi program się porusza nie zostanie sprawdzona.

Wszystkie przedstawione wyżej uwagi dotyczą porównań z dBASE III plus. Nowa wersja programu dBASE — dBASE IV zawiera wiele z omówionych powyżej rozszerzeń.

Za zasadniczy powód powodzenia kompilatora Clipper '87 należy jednak uznać fakt iż pozwala on otrzymywać programy w znacznie szybszej niż interpretowana wersji wykonywalnej (**.exe**). Dodatkowo uwalnia to końcowego użytkownika od ścisłego powiązania ze środowiskiem dBASE i konieczności częściowego przynajmniej poznania go, a także zmniejsza ryzyko ukradzenia części kodu źródłowego — w dBASE III plus musiały być udostępniane wersje źródłowe, co może prowadzić do nadużyć.

Sam kompilator, a także otrzymywane za jego pomocą programy nie mają specjalnych wymagań sprzętowych. Wystarczy im komputer klasy XT wyposażony w 640 KB pamięci operacyjnej i dwie stacje dyskietek, choć oczywiście użycie dysku twardego znacznie przyśpiesza działanie samego kompilatora jak i przygotowanych za jego pomocą programów. W zależności od tego, czy pisany program ma pracować w sieci, czy na jednym komputerze, potrzebny będzie MS-DOS (PC-DOS) 2.xx lub 3.xx. Ostatnim wymogiem jaki musi być spełniony, jest zapewnienie w pliku konfiguracyjnym możliwości otwarcia naraz 20 plików i korzystania z 8 buforów.

Clipper jest kompilatorem wsadowym, nie posiadającym zintegrowanego środowiska typu Turbo Pascala. Otrzymanie programu wynikowego wymaga wykonania kompilacji (do postaci **.obj**) i łączenia podprogramów bibliotecznych. Clipper wyposażony jest we własny program łączący (konsolidator) PLINK86, jednak jest on bardzo wolny, toteż najczęściej korzysta się z dostępnego w DOS-ie konsolidatora LINK, lub jeszcze szybszego TLINK (dostarczanego razem z Turbo C i Turbo Asemblerem Borlanda). Wszystkie standardowe procedury i funkcje języka zawarte są w dwóch bibliotekach — CLIPPER.LIB i EXTEND.LIB, jednak oprócz nich istnieją biblioteki napisane w C i asemblerze, pozwalające na szersze wykorzystanie możliwości komputera na przykład do tworzenia grafiki.

Kod wynikowy produkowany przez Clippera jest zawsze duży — dla trywialnych programów, składających się z jednej instrukcji, zajmuje on około 100—110 KB. Dalszy wzrost rozmiaru programu nie jest już szybki, ale i tak sensowny, niezbyt skomplikowany program zajmuje zwykle minimum 130 KB. Jest to jedna z najmniej wygodnych cech Clippera — duże pliki są trudniejsze w obsłudze, wymagają więcej czasu podczas ich ładowania do pamięci operacyjnej, i — co również ważne — zmniejszają dostępną pamięć operacyjną dla wołanych komendą RUN (lub !) innych programów i poleceń DOS-u.

Clipper pozwala na dołączenie do kodu procedur pisanych w C i asemblerze, i przygotowanych w postaci **.obj**. Poprawne współdziałanie tak przygotowanych podprogramów wymaga zadeklarowania ich w programie głównym jako EXTERNAL, i wykorzystania podczas kompilacji procedur plików extend.h i extend.mac, zawierających zestawy potrzebnych definicji.

Niezwykle cenną rzeczą jest istnienie w Clipperze debuggera. Jego wykorzystanie wymaga włączenia do przygotowywanego programu pliku DEBUG.OBJ. W tak przygotowanym programie można w każdej chwili wywołać debugger klawiszami alt-d, lub na drodze programowej, za pomocą funkcji ALTD(). Debugger wyposażony jest we własne menu pozwalające na kontrolowanie wartości zmiennych, wyrażeń, otwartych plików, a także na krokową pracę programu.

Razem z kompilatorem dostarczane są różne programy pomocnicze w wersji źródłowej — z jednej strony pozwalają one na wykonanie pewnych prostych a przydatnych operacji, na przykład tworzenia, edycji i indeksowania plików **.dbf** (program DBU), z drugiej zaś strony mogą służyć jako przykłady, ułatwiające zrozumienie zasad posługiwania się systemem.

Wersja kompilatora Clipper Summer '87 nie jest już najnowszą, oferowaną przez Nantucket. Najnowsza wersja ma numer 5.0, i przyniosła dalsze rozszerzenia języka (np. możliwość korzystania z tablic wielowymiarowych, dalsze funkcje sieciowe) oraz eliminację wielu błędów, utrudniających korzystanie z kompilatora.

Zdzisław Marchel



Nazwę dBase znają wszyscy, nawet jeśli nigdy nie korzystali z samego programu. Nie wszyscy jednak wiedzą, że dBase ma kilku bardzo groźnych konkurentów — przy pełnej kompatybilności z dBase znacznie szybszych i wygodniejszych w użyciu. Tymi konkurentami są produkty firmy Fox Software.

Cała historia zaczęła się w 1983 roku, kiedy niejakiemu Davidowi Fultonowi zaproponowano przygotowanie dużej bazy danych dla którejś z fabryk samochodów w Detroit. Fulton po kilku tygodniach pracy doszedł do wniosku, że jedynym narzędziem nadającym się do realizacji zadania, które przed nim stało, mogłaby być dBase II. Było jednak kilka małych „ale" — program był za wolny, i brakowało w nim kilku potrzebnych poleceń. Fulton porozumiał się z firmą Ashton-Tate (producentem dBase) i otrzymał ich zgodę na prace nad usprawnieniem dBase. Program który dzięki temu powstał w 1984 roku, to pierwsza wersja FoxBASE. Później była FoxBASE+, i jej wersje na różne komputery — 386, MacIntosh i maszyny Unix-owe, a w 1989 pojawił się program FoxGraph i pierwsza wersja FoxPro.

O ile FoxBASE to niemal wierna kopia dBase, różniąca się głównie szybkością (czasem wyższą nawet niż programy kompilowane za pomocą Clipper-a), o tyle FoxPro to już zupełnie nowa jakość. Najkrócej można FoxPro opisać następująco: weź dBase, przyśpiesz go kilka razy, dołóż rozwijane menu, możliwość pracy w kilku oknach i obsługę za pomocą myszy. To co powstało popraw tak, żeby było jeszcze wygodniejsze w użyciu, rozbudowując zestaw dostępnych instrukcji i poleceń. FoxPro 1.02 jest już gotowy.

Nieco dokładniej na temat tego programu wypowiedziała się kilka miesięcy temu nasza konkurencja, nie będę

więc go dokładnie opisywać — spróbuję tylko dokonać pewnego podsumowania. FoxPro pozwala na wykonywanie tych samych operacji, które były dostępne w dBase, jednak dzięki znacznie lepszemu interfejsowi użytkownika uzyskanie tych samych efektów jest znacznie szybsze i łatwiejsze. Tak to wygląda podczas pracy pojedynczego użytkownika, jednak FoxPro znakomicie nadaje się na narzędzie do pisania gotowych programów oprócz wygodnych narzędzi do generowania raportów i wydruków mamy bowiem do dyspozycji porządny debugger, możliwość napisania własnego help-a (obsługiwanego przez FoxPro) i tak dalej. W skład pakietu wchodzi także program przygotowujący dokumentację. Nie ma wprawdzie możliwości przygotowania samodzielnego pliku *.exe, tak jak jest to możliwe w przypadku Clipper-a, jednak do wersji podstawowej FoxPro można dokupić bibliotekę *runtime*, którą można następnie bez żadnych ograniczeń rozprowadzać z przygotowanymi przez siebie (oczywiście za pomocą legalnej kopii FoxPro) aplikacjami. Na rynku szanującym legalność oprogramowania takie rozwiązanie jest równie akceptowalne, jak pojedynczy plik *.exe.

Powyższe uwagi dotyczą FoxPro 1.02, tymczasem w momencie gdy będziecie czytać te słowa, powinna być w sprzedaży jej następczyni, FoxPro 2.0. Jeśli napiszę, że nowa wersja ma się do starej tak, jak stara do dBase, nie popełnię chyba wielkiego błędu. Składa się na to kilka czynników.

Z punktu widzenia końcowego użytkownika gotowej aplikacji najważniejsza będzie różnica w szybkości. Dzięki zastosowaniu nowej technologii (oczekującej na przyznanie patentu) o nazwie Rushmore, szybkość dostępu do baz danych jest około sto razy większa, niż w poprzedniej wersji programu (która i tak biła całą konkurencję o głowę). Wybranie pasujących do określonego warunku rekordów z bazy danych liczącej milion rekordów zajmuje kilka dziesiątych sekundy! Rushmore opiera się na pełnym indeksowaniu zbioru (według wszystkich możliwych pól), co jednak samo w sobie nie tłumaczy tak dużej szybkości. Podczas pokazu w Warszawskiej Victorii chwilami czuć było zapach siarki i smoły, co może sugerować konszachty Fox-a z siłami nieczystymi. Może zresztą gdzieś w okolicy smołowano jakiś dach. Niezależnie od tego jakie siły leżą u jej podnóża, szybkość jest po prostu niewiarygodna. Co ważne, potrzebne indeksowanie odbywa się w sposób automatyczny i jest zupełnie przeźroczyste dla użytkownika, który nie musi nic o nim wiedzieć. Pełne indeksowanie może oznaczać konieczność trzymania na dysku olbrzymich indeksów — tak jednak nie jest, gdyż dzięki zastosowaniu odpowiedniego formatu, indeksy udało się zmniejszyć o kilkadziesiąt (do 80) procent. Jest to zresztą o tyle logiczne, że im większy jest indeks, tym wolniejszy dostęp do informacji w nim zawartych.

Umysł ludzki, a w szczególności umysł osobnika piszącego o oprogramowaniu, ma tendencję do wyszukiwania dziury w całym — tak też było tym razem. Udało się ustalić, że Rushmore zawodzi, gdy poszukiwanie dotyczy nie całkowitej zawartości pola rekordu, a jego części — kto jednak szuka w bazie danych informacji na temat liczby mieszkańców miasta stołecznego W. mających w swoim nazwisku ciąg liter „tęs"? Zwykle chodzi o poszukiwania dotyczące zawartości całego pola, lub jego pierwszych znaków, a wtedy Rushmore pokazuje swój lwi pazur.

Szybkość, choć niezwykle istotna, nie jest (przynajmniej dla programistów) parametrem najważniejszym. Programista potrzebuje wygodnych i wydajnych narzędzi do przygotowywania różnych fragmentów programów. FoxPro 2.0 jest pod tym względem starannie dopracowane — wszystkie formularze i raporty można zaprojektować interakcyjnie, korzystając z myszy, a FoxPro wygeneruje odpowiedni fragment programu automatycznie. W trakcie projektowania ekranów mamy do dyspozycji kilka rodzajów pól, guzików i innych elementów, obsługiwanych za pomocą klawiatury lub myszy. W najprostszym wariancie wszystkie pola rekordu znajdą się automatycznie na ekranie, do programisty będzie tylko należało ewentualne rozmieszczenie ich w odpowiednich miejscach. Można jednak dołożyć pola odpowiadające różnym zmiennym, tudzież sterujące działaniem programu — np. skoki do początku, końca bazy, lub przejścia do edycji następnego, poprzedniego rekordu.

Co więcej, menu programu jest w pełni konfigurowalne, toteż można z niego wyrzucić te elementy, które nie będą potrzebne użytkownikowi — z jednej strony ułatwi to korzystanie z gotowego produktu, z drugiej zaś zmniejszy ryzyko wykonania operacji, która mogłaby być niebezpieczna dla samej bazy danych.

FoxPro ma jeszcze jedną zaletę. Jest w nim możliwe sortowanie zgodne z polskim alfabetem, można bowiem zdefiniować ciąg kodów ASCII, według którego wykonywane będzie sortowanie. Nie oznacza to jeszcze pełnej polonizacji pakietu, ale według informacji od polskiego dystrybutora jest ona możliwa (łącznie z polskim menu i komunikatami o błędach).

Trudno na kilku stronach maszynopisu dokładnie przedstawić tak rozbudowany produkt, jak FoxPro 2.0, toteż tekst ten należy traktować raczej jako pewien sygnał, niż opis programu. Nie napisałem ani słowa o SQL, o możliwości pisania własnych procedur w C i asemblerze, o wykorzystaniu dodatkowej (*expanded*) pamięci, o pracy w innych trybach tekstowych niż podstawowy (25*80), o możliwości przygotowywania programów w postaci projektów (z automatycznym kompilowaniem potrzebnych fragmentów programu, znajdujących się w różnych plikach; decyzja o kompilacji uzależniona jest od daty ostatniej modyfikacji) i wielu innych rzeczach.

Nie ma jednak róży bez kolców. FoxPro jest wyraźnie droższy od dBase i Clipper-a. Poprzednia wersja (FoxPro 1.02) kosztowała pod koniec maja 9.8 mln zł (plus 7.5 mln biblioteka *runtime*), ile będzie kosztować 2.0, jeszcze nie wiadomo, pewnym jest, że nie będzie tańsza. Biorąc jednak pod uwagę korzyści, jakie można odnieść inwestując w FoxPro, różnica w cenie między pakietem a jego konkurencją wydaje się uzasadniona.

Marcin Borkowski

FileFix
z pakietu Norton Utilities 5.0

Rok temu pojawiła się kolejna wersja programów narzędziowych firmy Norton Computing. Pakiet Norton Utilities 5.0 jest kontynuacją poprzednich wersji. Zmieniono w nim jednak szatę graficzną, a także zmodyfikowano programy z wcześniejszych wersji i dodano nowe.

Jedną z nowości jest FileFix — program do odzyskiwania i porządkowania uszkodzonych danych z plików tworzonych przez Lotusa 1-2-3, Symphony i dBASE. FileFix najlepiej prezentuje swoje możliwości przy odzyskiwaniu baz danych utworzonych przez dBASE i przez programy napisane pod Clipperem.

Do odtwarzania informacji z pliku wykorzystywane są informacje zapisane w nagłówku pliku. Zawierają one numer wersji dBASE użytej do stworzenia pliku danych, adres pierwszego rekordu danych, łączną długość rekordu i kolejno: nazwy, typy i długości poszczególnych pól.

Najpierw należy wybrać plik, który chcemy naprawić oraz plik roboczy, do którego będą zapisywane odzyskane dane.

Dostępne są trzy metody odzyskiwania danych:

— automatyczna — Zalecana na początek i najczęściej wystarczająca, jeśli nagłówek opisujący strukturę rekordów jest nie uszkodzony. Bardzo pomocna przy długich bazach danych.

— półautomatyczna — Przydatna w przypadku gdy pierwszą metodą nie udało się odzyskać większości lub najważniejszych danych. FileFix przerywa odzyskiwanie danych po wykryciu uszkodzonego rekordu, pozwala go poprawić, dołączyć do już odzyskanych rekordów i kontynuować przeglądanie.

— „ręczna" — Bardzo czasochłonna, ale jedyna skuteczna dla bardzo uszkodzonych baz np. po odzyskaniu pliku przez program CHKDSK. Polega na oglądaniu, poprawianiu i oddzielnym dołączaniu każdego rekordu.

Dodatkowo można użyć bardziej zaawansowanych opcji:

— limity pól dla plików tworzonych przez Clipper — Włączenie tej opcji dopuszcza większą liczbę pól i dłuższe pola niż tworzone przez dBASE.

— zaniechanie sprawdzania przemieszczenia danych względem ich pól — Włączenie tej opcji ma sens gdy chcemy tylko sprawdzić, czy pola nie zawierają niewłaściwych znaków np. EOF, NULL.

— dokładne sprawdzanie znaków w polach typu CHARACTER — Włączenie tej opcji powoduje, że w polach typu CHARACTER każdy znak, którego kod ASCII jest większy niż 128, zostanie zamieniony na spację.

Po odpowiednim ustawieniu powyższych opcji, automatycznym teście poprawności zapisu pliku na dysku i wstępnym teście nagłówka można rozpocząć odzyskiwanie danych lub przejrzeć opis struktury rekordu.

Przejrzenie nagłówka jest najważniejszym etapem odzyskiwania danych. W razie wykrycia pewnych niezgodności można poprawić opis rekordu poprzez edycję jego pól, albo lepiej, poprzez wczytanie poprawnego nagłówka z innego pliku o tej samej strukturze pól np. ze starszej wersji bazy danych.

Warto upewnić się, czy nagłówek jest poprawny, ponieważ kolejne rekordy są oddzielone tylko spacją i nie ma pomiędzy nimi żadnych znaczników, sum kontrolnych itp. Dlatego błędny nagłówek praktycznie uniemożliwia odzyskanie danych. FileFix pomaga ustalić jaka jest długość rekordu, a następnie można dowolnie zmieniać każde jego pole, aż do uzyskania zadawalającego rezultatu.

Gdy nagłówek jest poprawny, włączamy procedurę odzyskiwania rekordów danych, którą można w każdej chwili przerwać, aby na przykład zmienić metodę odzyskiwania danych. Jeśli w tym momencie konieczne będą dodatkowe poprawki w nagłówku, można powrócić na początek całej procedury. Na zakończenie pracy program oferuje wydruk krótkiego raportu.

Po odtworzeniu bazy danych warto ją dokładnie przejrzeć. — Jeśli wynik pracy FileFix okaże się niewystarczający trzeba spróbować jeszcze raz uruchamiając program z innymi opcjami.

Program FileFix jest prosty w obsłudze (obsługuje mysz, dokładnie opisuje każdy wykonywany etap). Może przynieść nieocenione usługi przy odtwarzaniu popsutych baz danych, ale nie zawsze da sobie radę z wszystkimi błędami. Bardziej złośliwe przypadki (błędny nagłówek, zniszczone sektory w pliku) należy poprawiać ręcznie, co wymaga pewnej wprawy i wiedzy o strukturze ratowanego pliku.

Marek Sawicki

ALT-286
jeszcze raz

W nawiązaniu do testu sprzedawanego przez nas komputera, który ukazał się w czerwcowym numerze Bajtka, chciałbym przedstawić następujące uwagi:

1. **Problem z mechaniczną konstrukcją ekranu występował tylko w pierwszej dostawie komputerów, od następnej przestał istnieć.**
2. **Fakt, iż sprzedawany przez nas model komputera jest oferowany również przez inne firmy, powinien raczej dobrze świadczyć o samym komputerze — jest na niego bardzo wysoki popyt, jest to jeden z najmocniej reklamowanych komputerów w piśmie ASIAN SOURCES. W tekście testu wzmianka o „braku markowości" komputera ma zdecydowanie negatywny charakter. Zasada stosowania własnych etykiet jest bardzo powszechna w komputerach sprowadzanych z Tajwanu — od następnej dostawy jest to w naszym przypadku nazwa KAM.**
3. **Obecnie jest już dostępna również wersja 386SX 20 MHz z 32k CACHE MEMORY.**

Marek Górecki
BH KAM-PL

Staraliśmy się możliwie dokładnie opisać testowany egzemplarz komputera, niemożliwe więc było umieszczenie w teście informacji na temat zachowania się innych egzemplarzy. Pojęcie markowości sprzętu jest dość płynne, dla mnie sprzęt markowy to taki, na którym podpisał się jego producent, a nie sprzedawca. Do komputera wrócimy — wczesną jesienią planuję napisać kilka słów o jego dalszej, po zamknięciu testu, eksploatacji.

(mb)

„PRZYWIEZIONE z CeBIT-u"

ŁATWE SZUKANIE

W jednym z najmniejszych stoisk, jakie istniały na targach, dostałem do ręki demonstracyjną wersję słownika Webstera.

Kiedy już miałem odchodzić sympatyczna Amerykanka z którą rozmawiałem powiedziała — „Poczekaj jeszcze chwilę, myślę, że mam jeszcze coś ciekawego dla Ciebie!" Po kilku próbach otwarcia zatrzaśniętej szafki dostałem do ręki kolejną dyskietkę i odbitkę kserograficzną jakiegoś artykułu — „Obejrzyj to dokładnie, to jest naprawdę interesująca rzecz!"

Jakimś dziwnym trafem dyskietka i jej opis przyjechały do Warszawy osobno. Kiedy wziąłem się za porządkowanie przywiezionych materiałów, trafiłem na samą dyskietkę, i przyznam, że nie bardzo potrafiłem sobie przypomnieć, skąd ją mam. Byłem gotów odzyskać przy pomocy **del *.*** nośnik, ale ostatecznie — zanim to zrobiłem — postanowiłem obejrzeć zawartość. Po kilku minutach prób z wypiekami na twarzy zacząłem przekopywać się przez kilogramy prospektów, w poszukiwaniu tej jednej odbitej na ksero stroniczki.

Co zrobiło na mnie takie wrażenie? Podeprę się przykładem. Wyobraźcie sobie, że dysponujecie bazą danych (dBase), zawierającą spis wszystkich byłych uczniów waszej szkoły. Przychodzi do was ktoś, i mówi, że potrzebny mu jest adres Janka Ziółkowskiego. Jak go znaleźć? Można spróbować za pomocą LOCATE FOR, ale nazwiska potrafią płatać figle — czy to nie był przypadkiem Ziułkoski, albo jeszcze jakoś inaczej? A czy ten Janek to od Janusza, czy od Jana? Widzicie na czym polega problem?

Otóż na dyskietce, której historię opisałem powyżej, znajdował się program Friendly Finder, firmy Proximity Technology, który potrafił sobie dać radę w takich właśnie sytuacjach. FF ma dwie wersje — jedna jest programem rezydentnym, z którego można korzystać w dowolnym momencie, niezależnie od tego, co wykonuje się jako podstawowa aplikacja, druga to procedura biblioteczna, łatwo dołączalna do Clippera. Algorytym poszukiwań dopasowany jest do języka angielskiego i opiera się na różnych bliżej niesprecyzowanych w opisie regułach, które pozwalają na wykrycie słów podobnych brzmieniowo — oczywiście po angielsku. W jednej z kilku przykładowych baz danych znajdują się dane wszystkich (?) amerykańskich miast. Próbowałem różnych trików — np. poszukiwania Waszyngtonu pisanego Uoshinton (co fonetycznie powinno dać efekt zbliżony do oryginalnej pisowni) — i FF dawał sobie radę bezbłędnie.

Podobne techniki można zastosować w stosunku do języka polskiego i choć nie jest to program pierwszej potrzeby, na pewno znalazłby wielu nabywców.

Marcin Borkowski

PORÓWNYWANIE DANYCH

Jak skopiować pliki i dyskietki za pomocą poleceń copy i diskcopy już wiecie. Warto by jednak było móc jeszcze sprawdzić, czy podczas kopiowania nie został popełniony jakiś błąd. Ktoś kiedyś powiedział, że dopóki częstotliwość występowania błędów daje się zmierzyć, jest na pewno za duża. Jest w tym stwierdzeniu trochę przesady, ale tylko trochę — stosowane obecnie dyski i stacje dyskietek nie są stuprocentowo pewne. Błędy zdarzają się niezwykle rzadko, ale gdy potrzebny zbiór danych nie daje się odczytać, świadomość że zdarzyło się coś niezwykle mało prawdopodobnego jest małą pociechą.

DOS pozwala na kontrolowanie poprawności zapisu na kilku poziomach. Pierwszy z nich to polecenie verify — zmuszające system operacyjny do sprawdzania poprawności każdej wykonanej operacji zapisu na dyskietce. Polecenie to ma postać verify=on, lub verify=off, w zależności od tego czy chcemy DOS zmusić do sprawdzania, czy do zaprzestania sprawdzania poprawności zapisu. Jeżeli nie pamiętamy, jakie polecenie było wydane ostatnio, wystarczy napisać samo verify, by uzyskać na ekranie tę informację. Sprawdzanie poprawności zapisu jest dość czasochłonne, i może wyraźnie spowolnić działanie komputera.

Wbrew temu, co może się Wam wydawać, sprawdzenie poprawności zapisu nie oznacza wcale porównywania zapisanej na dysku informacji z tą, która miała się na nim znaleźć. DOS sprawdza jedynie, czy w zapisanym sektorze nie występuje błąd parzystości — ostatni bajt jest zawsze pewną funkcją wszystkich pozostałych bajtów, co pozwala na prostą i dość skuteczną kontrolę poprawności zapisu i odczytu (stosowaną praktycznie we wszystkich komputerach). Teoretycznie może się jednak zdarzyć, że wystąpi błąd, który przez zwykłą kontrolę parzystości nie zostanie wykryty. Aby się przed tym uchronić, trzeba skorzystać z mniej wyrafinowanego sposobu postępowania, jakim jest porównanie dwóch plików bajt po bajcie.

Służy do tego celu polecenie comp, potrzebujące dwóch parametrów, określających co z czym ma być porównywane. Przykładowo:

comp *.doc a: — porównane zostaną ze sobą wszystkie pliki o rozszerzeniu doc znajdujące się w głównym katalogu dyskietki a: z plikami o tych samych nazwach, znajdującymi się w katalogu aktualnym.

Jeśli nie podamy drugiego parametru DOS zapyta o nazwę drugiego pliku, lub nazwę dysku — naciśnięcie samego klawisza Enter spowoduje przyjęcie jako drugiego parametru aktualnego katalogu, toteż polecenie **comp *.doc a:** może również przyjąć formę **comp a:*.doc.** Zwykle jednak nie warto stosować takich sztuczek, które łatwo prowadzą do omyłek.

DOS porównuje ze sobą pliki mające te same nazwy, i identyczne długości — plików o różnych długościach nie da się porównać za pomocą polecenia comp. Jeżeli pliki nie są identyczne, DOS wypisze dziesięć pierwszych różnic (podając adres względem początku pliku, oraz heksadecymalną zawartość bajtu w obu plikach), i przerwie porównywanie.

W ten sposób można porównać ze sobą pojedyncze pliki, co jednak zrobić, gdy kopiowana była cała dyskietka? Tu też można się posłużyć poleceniem DOS-u — diskcomp. Działa ono dokładnie tak samo jak polecenie diskcopy — należy podać nazwy dwóch stacji, w których znajdują się dyskietki do porównania (w ostateczności może to być ta sama nazwa, będzie wtedy trzeba żonglować dyskietkami w takt poleceń systemu operacyjnego).

Marcin Borkowski

Z portu do portu...

Bardzo często się zdarza, że kupienie nowego sprzętu powoduje lawinowe zapotrzebowanie na różne drobiazgi.

Tak też stało się tym razem — po zakupie drukarki laserowej. Została ona początkowo podłączona do portu lpt2:, żeby umożliwić korzystanie (przynajmniej przez jakiś czas) ze starej LC-200. I wszystko byłoby dobrze, gdyby każdy używany przeze mnie program pozwalał na łatwą zmianę konfiguracji — niestety, wiele programów z uporem godnym lepszej sprawy uważa, że drukarka jest podłączona do lpt1:, i nie daje im się w żaden sposób przemówić do kodu maszynowego. Rozwiązania trzeba więc szukać na zewnątrz, poza programem.

Zadanie można sformułować tak — potrzebny jest program, który pozwoli na zamianę przypisania portów — to znaczy zamieni lpt1: na lpt2: i odwrotnie, oczywiście bez przełączania kabelków. Dla kogoś nie znającego dobrze PC-eta zadanie może się wydawać trudne, i w pewnym sensie jest — wymaga bowiem sięgnięcia do podstaw BIOS-u.

Wszystkie urządzenia wejścia-wyjścia (porty szeregowe i równoległe, stacje dyskietek i twardy dysk itd.) są przez procesor obsługiwane za pomocą instrukcji IN i OUT. W odróżnieniu jednak od niektórych prostych komputerów ośmiobitowych, adresy pod którymi należy szukać różnych urządzeń zewnętrznych nie są ustalone raz na zawsze, a zapamiętane w obszarze zmiennych systemowych BIOS-u. Ponieważ pierwsza i druga drukarka są różnymi urządzeniami, mają różne adresy I/O i wystarczy zamienić je ze sobą, by uzyskać pożądany efekt. Adresy I/O drukarek (IBM PC pozwala teoretycznie na podłączenie czterech różnych naraz, choć wymaga to posiadania odpowiedniej karty z kilkoma interfejsami równoległymi) są zapamiętane w ośmiu bajtach poczynając od adresu 0:408H (każdy adres I/O jest dwubajtowy).

Do napisania programu najlepiej nadaje się asembler, choć można skorzystać nawet z BASIC-a*, nie wspominając o takich językach jak Pascal i C. Gotowe rozwiązanie (bardzo oszczędne, bo nie drukujące informacji o tym co zostało zrobione) jest przedstawione na wydruku. Przygotowanie postaci swap.com wymaga wykonania (po wpisaniu listingu i nadaniu mu nazwy swap.asm) następujących operacji:

tasm swap.asm ; lub masm swap.asm, zależnie od posiadanego asemblera
tlink swap ; lub link swap
exe2bin swap.exe swap.com

i skasowania niepotrzebnych już plików — wszystkich poza swap.com (i ewentualnie swap.asm). Od tego momentu można korzystać z nowego polecenia — jego pierwsze wydanie zamieni lpt1: na lpt2: i odwrotnie, powtórne odtworzy sytuację początkową.

```
cseg        segment
assume      cs:cseg

start:      xor     ax,ax
            mov     ds,ax
            mov     ax,ds:[408h]
            mov     bx,ds:[40Ah]
            mov     ds:[408h],bx
            mov     ds:[40Ah],ax
            mov     ax,4C00h
            int     21h

cseg        ends
end         start
```

Marcin Borkowski

* patrz Bajtek 3/89, str. 25

TERMINAL CP/M

O tym, że stacja dysków FDD 3000 umożliwiła użytkownikom ZX Spectrum pracę w systemie CP/M wiedzą już zapewnie wszyscy.

Co więc jest przyczyną niewielkiej jego popularności? Zapewne brak dostatecznej ilości oprogramowania i nieznajomość jego pracy. W kilku artykułach postaramy się przybliżyć nieco konstrukcję i pracę systemu, a także przedstawić sposoby rozszerzania jego możliwości.

Aby na dowolnym komputerze można było zainstalować system CP/M, musi on posiadać następujące cechy sprzętowe: procesor 8080 lub Z80, pamięć RAM-minimum 24KB-umieszczoną od adresu 0, stacja dysków i terminal alfanumeryczny. W ZX Spectrum pamięć RAM nie rozpoczyna się od adresu 0 i to jest już dostateczna przyczyna, dla której konstruktorzy musieli szukać innego rozwiązania. Jest ono dość niekonwencjonalne. Stację dysków FDD 3000 wyposażono we własny procesor Z80 (zegar 4MHz) i 64 KB pamięci RAM. Jest to zatem samodzielny mikrokomputer, o możliwościach przetwarzania większych od Spectrum. Do pełnej realizacji systemu zabrakło jedynie terminala, ale do tego zadania świetnie nadaje się Spectrum.

Przy pracy w systemie CP/M podstawowym procesorem (realizującym program) jest więc ten zawarty w stacji. Komunikacja z ZX Spectrum następuje poprzez szybkie dwukierunkowe łącze równoległe (niestety tylko 4-bitowe, przez co transmisja jednego bajtu następuje w dwóch krokach). Łączem tym przesyłane są dane do terminala i z powrotem.

Terminal współpracujący z FDD 3000 ma za zadanie wyświetlanie na ekranie monitora danych przychodzących z systemu i wysyłanie do niego znaków wpisywanych z klawiatury. ZX Spectrum w roli terminala nie jest więc zbytnio obciążony pracą, co pozwala wykorzystać go również do innych zadań, rozszerzających możliwości systemu. Taki podział zadań jest przyczyną skomplikowanego uruchamiania CP/M-u. Na początku trzeba wczytać do ZX Spectrum z dyskietki TOS program realizujący funkcję terminala i następnie po wymianie dyskietki uruchomić CP/M. Warto wspomnieć, że firma TIMEX produkowała również specjalny terminal, przeznaczony wyłącznie do pracy pod CP/M-em.

Istnieją zasadniczo dwa programy realizujące funkcje terminala. Jeden z nich opracowała firma TIMEX i użytkownik otrzymuje go wraz ze stacją FDD 3000. Występuje on w dwóch wersjach. Pierwsza jest przeznaczona dla ZX Spectrum. Terminal ten wyświetla znaki o kiepskiej czytelności (matryca 4*8), co jest związane z trudnością realizacji 64 kolumn tekstu na tym komputerze. Podobnie działają popularne edytory tekstu. Druga wersja, przeznaczona dla komputera TIMEX dzięki dodatkowym jego możliwościom graficznym zapewnia znakomitą czytelność tekstu. Drugi program terminala został opracowany w firmie Polbrit i do niedawna był sprzedawany przez CSH. Obecnie można go kupić na giełdzie, co jest nielegalne. Jest on przeznaczony głównie do ZX Spectrum. Nie opracowano niestety wersji dla komputera TIMEX. Jest to szczególnie przykre, gdyż posiada on wiele zalet i funkcjonalnością znacznie góruje nad produktami programowymi firmy TIMEX.

Oprócz zwykłego wyświetlania znaków na ekranie (zapewnianego przez każdy terminal) do prawidłowej pracy wymagane jest oprócz tego realizowanie szeregu dodatkowych funkcji. Należą do nich między innymi kasowanie zawartości ekranu, ustawianie kursora, ustawianie atrybutów znaków itp. Wywołanie takich funkcji następuje poprzez wysłanie do terminala specjalnych sekwencji znaków. Sekwencje te nie są zobrazowywane na ekranie, służą tylko do uaktywnienia wspomnianych funkcji. Typowa sekwencja składa się z znaku ESC (znacznika sekwencji) i kilku parametrów. Ich wysyłanie jest analogiczne do zwykłego wypisywania znaków i jest dokonywane za pomocą zwykłych instrukcji PRINT lub WRITE.

Liczba realizowanych funkcji zależy od konkretnego terminala. Terminale TIMEX i POLBRIT znacznie się tutaj różnią. Pierwszy z nich realizuje jedynie podstawowe funkcje jak przesuwanie i ustawianie kursora w zadanym miejscu ekranu, kasowanie jego zawartości, podświetlanie wypisywanych znaków. Jest to bardzo skromny zestaw, dodatkowo znana z tajemniczości firma TIMEX nie dostarcza wraz z programem żadnego spisu tych komend. O tym jakie kody sterujące przypisano tym sekwencjom, użytkownik musi dowiedzieć się eksperymentalnie metodą prób i błędów na przykład analizując działanie właściwie zainstalowanych programów.

Terminal firmy Polbrit realizuje oprócz zestawu funkcji programu TIMEX szereg dodatkowych, które znacząco podnoszą jego użyteczność. Umożliwia on między innymi pełną obsługę okien tekstowych, przesuwanie treści ekranu w dowolnym kierunku, obsługę grafiki i wiele innych. Program terminala zajmuje jedynie 7 KB pamięci RAM, pozostały wolny obszar może zostać wykorzystany do zapamiętywania treści kilku ekranów, do pamiętania danych (jako RAM dysk), a także jako bufor drukarki itp. Do zrealizowania tych możliwości użytkownik otrzymuje gotowe funkcje. Efekty graficzne sprowadzają się do rysowania okręgów, odcinków i punktów.

W tym odcinku ograniczymy się do omówienia wszystkich funkcji tego terminala, w następnym podamy sposoby na rozbudowę terminala o własne, niestandardowe funkcje użytkownika. Opiszemy również przykładowe rozszerzenia wykonane i wypróbowane przez nas.

Robert Magdziak
Maciej Pietraś
Jonasz Mayer

Literatura: 1. R. Magdziak, Centronics dla FDD 3000, Bajtek 7-8/90. 2. J. Mayer, Turbo Pascal — procedury graficzne na ZX Spectrum, Bajtek 11-12/90. 3. J. Mayer, Turbo Pascal — ramki semigraficzne. Bajtek 3/91.

SEKWENCJE STERUJĄCE TERMINALA POLBRIT

kod funkcji	*opis działania*
07	Wygenerowanie krótkiego dźwięku.
08	Skasowanie ostatniego znaku.
09	Przesunięcie kursora w prawo do kolejnej pozycji tabulacji.
ESC A	Przesunięcie kursora do góry o jeden wiersz.
ESC B	Przesunięcie kursora o jeden wiersz w dół.
ESC C	Przesunięcie kursora o jeden znak w prawo.
ESC D	Przesunięcie kursora o jeden znak w lewo.
ESC E	Skasowanie zawartości linii w której znajduje się kursor. Linie poniżej pozycji kursora przesuwane są o jeden wiersz w górę.
ESC F	Po wysłaniu tej sekwencji wszystkie znaki będą wypisywane w negatywie (inverse video).
ESC G	Przywrócenie normalnego trybu wyświetlania znaków (normal video).
ESC H	Przesunięcie kursora do lewego górnego rogu ekranu.
ESC J	Kasuje zawartość ekranu i umieszcza kursor w lewym górnym rogu.
ESC K	Wstawia pustą linię na wysokości aktualnej pozycji kursora. Wszystkie linie poniżej kursora przesuwane są o wiersz w dół.
ESC L	Kasuje zawartość linii w której znajduje się kursor, od jego pozycji do końca linii.
ESC M al ah dl dh	Przesłanie bloku bajtów z pamięci ZX Spectrum do FDD 3000, o długości dl+256*dh, od adresu al+256*ah w górę. Po wysłaniu sekwencji zamiast znaków z klawiatury odczytamy zawartość tego bloku.
ESC N al ah dl dh	Umieszczenie w pamięci ZX Spectrum bloku danych. Adres początku i długość identyczne jak dla „ESC M". Po wysłaniu sekwencji wysyłamy blok danych. Wykorzystanie tych dwóch sekwencji zostało opisane w [1] i [3].
ESC O xl xh yl yh rl rh	Narysowanie okręgu o środku w punktach: X=xl+256*xh Y=yl+256*yh i promieniu: R=rl+256*rh
ESC P xl xh yl yh	Zapalenie punktu o współrzędnych X, Y jak dla „ESC O".
ESC Q xl xh yl yh	Wykreślenie odcinka od ostatnio zapalonego punktu do punktu o zadanych współrzędnych X, Y (jak wyżej). Przykładowe wykorzystanie ostatnich 3 sekwencji zostało opisane w [2].
ESC T 1	Rozpoczęcie wyświetlania daty i czasu w górnym prawym rogu ekranu.
ESC T 0	Zaniechanie wyświetlania daty i czasu.
ESC U al ah	Wywołanie podprogramu w kodzie maszynowym i umieszczonym w pamięci pod adresem al+256*ah.
ESC V r m d g m s	Ustawienie zegara-kalendarza na żądany rok, miesiąc, dzień, godzinę, minutę i sekundę.
ESC X	Odczyt wskazania zegara-kalendarza. Dane są wysyłane w następującej kolejności: rok, miesiąc, dzień, dzień tygodnia, godzina, minuta, sekunda.
ESC Z	Przerwanie.pracy programu terminala.i powrót do BASiCa. Wznowienie pracy następuje poprzez USR 58859. Przykładowe zastosowanie opisano w [3].

FUNKCJE OPERUJĄCE NA TREŚCI CAŁYCH EKRANÓW

ESC I G nr	Wyświetla na ekranie zawartość ekranu o numerze nr (1..5).
ESC I P nr	Zapamiętanie w pamięci zawartości ekranu o numerze nr.
ESC I E nr	Wymienia zawartość ekranu aktualnie wyświetlanego z zawartością pamięci ekranu o numerze nr.

FUNKCJE OPERUJĄCE NA OKIENKACH TEKSTOWYCH

ESC W D nr kp lp lk ll	Zdefiniowanie okienka o numerze nr (1..9) i następujących parametrach: kp — kolumna początkowa (0..63) lp — linia początkowa (0..23) lk — liczba kolumn ll — liczba linii
ESC W S nr	Wybranie okienka nr jako aktualnego.
ESC S	Skasowanie zawartości aktualnego okienka od pozycji kursora w dół.
ESC R kr	Przesunięcie zawartości okienka w jednym z czterech kierunków kr: A na dół; B do góry; C w lewo; D w prawo.
ESC Y ln kl	Ustawienie kursora w obrębie okienka na pozycji: ln — linia i kl — kolumna.

```
Procedure POKEZXRAM(address:integer;value:byte);
begin
  write(#27'N',chr(lo(address)),chr(hi(address)),#1#0);
  write(chr(value));
end;
Function PEEKZXRAM(address:integer):byte;
var ch:char;
begin
write(#27'M',chr(lo(address)),chr(hi(address)),#1#0);
read(kbd,ch);
PEEKZXRAM:=ord(ch);
end;
```

Przykładowe użycie wymienionych sekwencji z poziomu Turbo Pascala ilustrują procedura PokeZXRam i funkcja PeekZXRam. Dokonują one zapisu lub odczytu pamięci ZX Spectrum.



SAM COUPÉ wraz z zasilaczem

SAM COUPÉ

W czasach kiedy komputery ośmiobitowe przechodzą do przeszłości, lansowanie ich jest zjawiskiem dziwnym, chyba że grafiką i dźwiękiem będą one równały się z Atari ST czy Amigą. Takim ośmiobitowym komputerem jest produkt angielskiej firmy MGT — SAM COUPÉ.

PIERWSZE WRAŻENIE

W środku estetycznego opakowania znaleźliśmy komputer i dość sporych rozmiarów zasilacz, który jest jednocześnie modulatorem sygnału wizyjnego. Poza tym w pudełku znajdował się kabel do połączenia komputera z magnetofonem, kaseta magnetofonowa, dyskietka z **DOS**em i programami użytkowymi, instrukcja użytkowania załączonego programu graficznego FLASH, oryginalna instrukcja użytkowania SAMa (User's Guide) oraz niezbyt udolnie zredagowane polskie tłumaczenie instrukcji.

Komputer ma bardzo nietypowy kształt. Posiada klawiaturę umiejscowioną około ośmiu centymetrów od przedniej krawędzi obudowy. Klawiatura jest nieco pochylona, co łącznie z miejscem na położenie nadgarstków przyczynia się do większego komfortu pracy. SAM COUPÉ wygląda solidnie i estetycznie.

ZALETY SAMa

- **grafika i dźwięk porównywalne z Amigą**
- **wbudowana stacja dysków**
- **dobry interpreter języka BASIC**
- **możliwość pracy w 80-kolumnowym trybie tekstowym**
- **duża pamięć RAM (256KB)**
- **zgodność programowa z ZX Spectrum**
- **łatwość rozbudowy**

PRÓBA URUCHOMIENIA

Od zasilacza odchodzą trzy dość długie przewody, które mają tendencję do ciągłego plątania się. Pierwszy z nich służy do połączenia komputera z odbiornikiem telewizyjnym, drugi należy połączyć z komputerem, a trzeci jest zakończony standardową wtyczką do gniazdka 220 V. Komputer ma z tyłu obudowy wiele gniazd, więc połączenie go z zasilaczem nie było sprawą oczywistą, pomimo tego, że wtyczka pasuje tylko do jednego gniazda (wetknięcie jej do innego jest fizycznie niemożliwe).

Dopóki nie uzyskamy obrazu na telewizorze bądź monitorze, nie możemy stwierdzić, czy SAM jest zasilany, bowiem ani na zasilaczu, ani w komputerze nie ma żadnej sygnalizacji tego faktu. Mała dioda podpisana „Power" bardzo by ułatwiła pracę z komputerem, zwłaszcza, że w SAMie istnieje mechanizm wyłączania wizji, gdy nikt przez dłuższy czas nie dotyka klawiatury. Bardzo łatwo zapomnieć, że komputer jest włączony.

Dużą zaletą SAMa COUPÉ jest możliwość korzystania z odbiornika telewizyjnego jak i z monitora. Telewizor łączymy przez zasilacz SAMa, a monitora możemy używać dzięki 21-stykowemu złączu SCART. Obraz na ekranie monitora jest wyraźny i kontrastowy. Jeśli używamy kolorowego monitora, to wystąpi problem dobrze znany wszystkim użytkownikom takich monitorów: teksty w trybie osiemdziesięcioznakowym będą słabo czytelne. Pewną niedogodnością w pracy z monitorem jest bardzo krótki przewód łączący z komputerem. Przez monitor słyszalny jest dźwięk z SAMa, lecz dodatkowo wśród wielu gniazd komputer posiada wyjście do wzmacniacza.

Przy uruchamianiu komputera powinna być pomocna broszurka w języku polskim. Niestety — jak na razie — stanowi ona tłumaczenie nie najlepiej wybranych fragmentów oryginalnej instrukcji.

Po podłączeniu komputera następuje pierwszy kontakt z jego klawiaturą. Jest ona wygodna i posiada **72 klawisze** (w tym 10 klawiszy funkcyjnych). Wzorem dla niej była klawiatura komputera Amstrad CPC 464. Wspominane już oparcie dla nadgarstków bardzo ułatwia obsługę. Klawiatura ma profesjonalny wygląd, chociaż w dotyku bardziej przypomina klawisze Amigi niż klawiaturę komputera klasy AT.

SAM ma z przodu obudowy miejsce na dwa napędy dyskietek **3.5 cala**. W modelu dostarczonym do redakcji był tylko jeden napęd, lecz zamontowanie kolejnego, dokupionego napędu nie wymaga żadnych zdolności. Wystarczy zdjąć zaślepkę, włożyć napęd i przykręcić dwie śrubki. Napędy są produkcji firmy Citizen i tylko one (ze względu na dołączenie specjalnego bufora) mogą być używane z SAMem COUPÉ. Stanowi to pewne ograniczenie, ponieważ nie można bezpośrednio podłączyć do SAMa napędu 5.25 cala.

SAM może współpracować również z dowolnym magnetofonem kasetowym. Standardowo sposób transmisji i zapisu danych jest podobny do ZX Spectrum. Istnieje jednak możliwość programowego ustawienia prędkości transmisji. Komputer przy wczytywaniu automatycznie rozpozna prędkość zapisu i ładowanie programu będzie przebiegało prawidłowo.

ZŁĄCZA I PRZYCISKI

Z tyłu komputera znajdują się dwa przyciski i jeden wyłącznik. Ten ostatni jest od zasilania a przyciski to **RESET** oraz **BREAK** (inicjalizacja przerwania niemaskowalnego). Oprócz tego znajdują się tam gniazda: **MIDI IN, MIDI OUT,** port joysticka, gniazdo dla myszy, 64-stykowe złącze, wyjście do magnetofonu, gniazdo dla świetlnego pióra będące zarazem gniazdem wyjścia dźwięku stereo do wzmacniacza, wspomniane wcześniej 21-stykowe gniazdo **SCART** oraz gniazdo zasilania.

Niewątpliwie, dla komputera klasy SAMa bardzo ważnym elementem jest **joystick**. Do otrzymanego egzemplarza komputera pasowały wszystkie posiadane przeze mnie joysticki. Te, które mają przełącznik dla różnych typów komputerów należy ustawić w położenie Amstrad CPC 464.

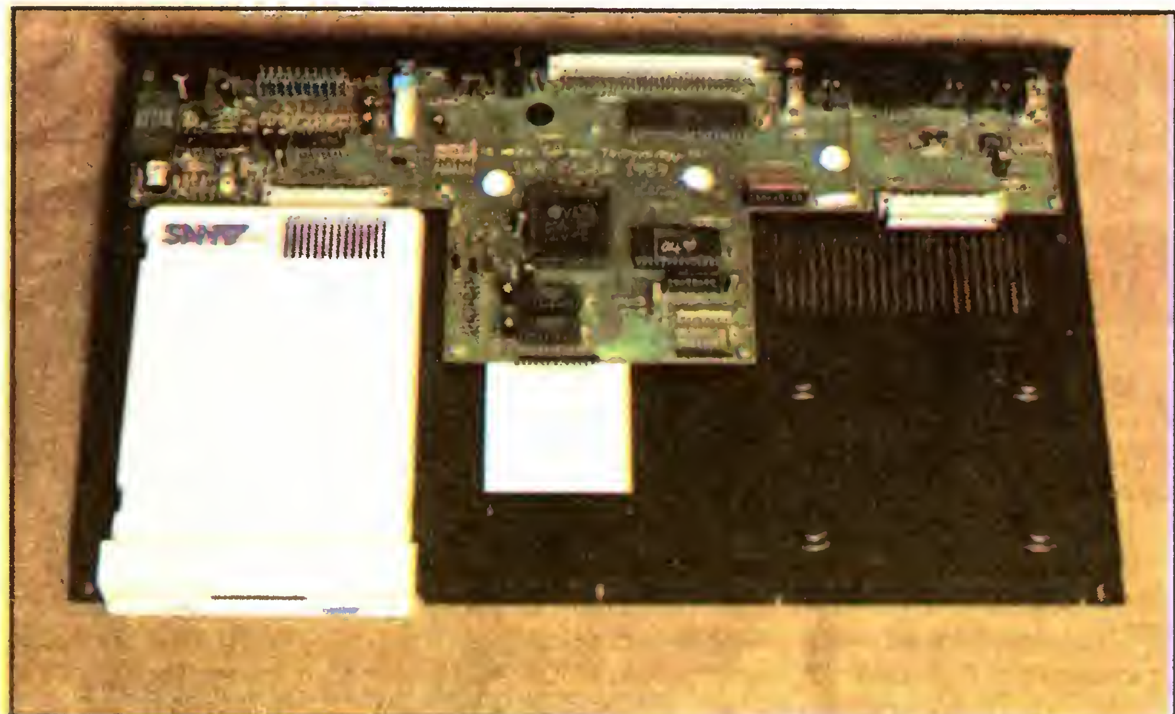
Niestety nie znalazłem w SAMie złącza standardu **Centronics** albo choć **RS-232**. Prawdopodobnie taka jest polityka firmy produkującej SAMa, że owe złącza — tak podstawowe dla użytkownika — dostępne są dopiero po zakupieniu odpowiedniego interface'u. Taka rozbudowa zestawu powoduje jednak zmniejszenie estetyki całości — wszyscy pamiętamy „sznur" interface'ów za ZX Spectrum...



Złącze i gniazda z tyłu obudowy ▲

ROZSZERZENIA

Co możemy podłączyć do SAMa? Prawie wszystko, jeśli wierzyć zapewnieniom producenta (patrz schemat zamieszczony dalej). Najbardziej odpowiednimi jednak zakupami będą: interface **Centronics** i **RS 232 C** oraz dodatkowa pamięć. Jeżeli chcemy, aby nasz SAM miał 512KB wystarczy nabyć odpowiedni moduł i bez rozbierania obudowy można go w ciągu dwóch minut zamontować. Pamięć może być rozbudowana maksymalnie do **4.5MB!**



◀ *Wewnątrz obudowy tylko osiem układów scalonych*

PROGRAMY

Oryginalnych, wykorzystujących możliwości SAMa programów jest mało, lecz ciągle powstają nowe. Na dyskietce (kasecie) dołączanej do zestawu jest kilka podstawowych, w tym programowy emulator ZX Spectrum, graficzny program **FLASH** oraz kilka programów demonstrujących wspaniałe (patrz tabelka) możliwości graficzne i dźwiękowe. Widziałem również profesjonalnie napisane gry, specjalnie dla SAMa COUPÉ, i muszę przyznać, że są one ładniejsze niż te same gry na Atari ST!

Producenci wiedzą, że komputer staje się popularny dzięki oprogramowaniu, więc wyposażyli SAMa w emulator na pewno ucieszy spectrumowych graczy. Istnieje specjalna przystawka o nazwie „Messenger" umożliwiająca uruchomienie każdego programu z ZX-a na SAMie, lecz jest to temat na inny artykuł.

SAM posiada wspaniały **BASIC** umożliwiający łatwe tworzenie własnych programów. W dialekcie tego języka znajduje się zestaw komend tworzenia sprite'ów, płynnego przesuwania dowolnej części ekranu, animacji, oraz oprawy dźwiękowej. Możliwe jest czytanie i zapisywanie dowolnego sektora dyskietki. **BASIC** SAMa jako jeden z nielicznych pozwala na używanie etykiet.

Najbardziej popularnym systemem dla komputerów ośmiobitowych jest



Wady SAMa

- **brak portów Centronics i RS-232**
- **nietypowe złącze napędu dyskowego**
- **źle przetłumaczona instrukcja obsługi**
- **brak sygnalizacji zasilania**

Spectrum. Jest to niepełna — ze względu na prawa autorskie — emulacja środowiska ZX Spectrum 48K. Pozwala ona na przenoszenie i uruchamianie z dyskietek SAMa około 90% oprogramowania ze Spectrum. W trybie Spectrum działa podłączony do SAMa joystick, co **CP/M**. SAM COUPÉ posiadający dużą pamięć oraz procesor Z80 ma wszelkie warunki, aby zainstalować na nim ten system. Posunięcie takie radykalnie by zwiększyło dostępność wszelkiego oprogramowania. Zalet korzystania z takich programów, jak: Turbo Pascal 3.0, dBase, Supercalc czy Wordstar nie trzeba chyba wymieniać.

PRZYSZŁOŚĆ

SAM COUPÉ jest zadziwiająco sprawnym, jak na ośmiobitowy, komputerem. Procesor **Z80B** o szybkości **6MHz**, sześciokanałowy generator dźwięku i dobra grafika stawiają go pomiędzy komputerami typu Commodore C-64 a Amigą. Jest on alternatywą dla tych użytkowników komputerów ośmiobitowych (np. Spectrum), których nie stać na zakup Amigi czy Atari ST, a chcieliby mieć — kompatybilny z popularnym w Polsce sprzętem — komputer wyposażony w stację dysków. Wspomniana wcześniej zgodność programowa z ZX Spectrum umiejscawia SAMa w potencjalnej czołówce najlepiej oprogramowanych maszyn.

Cena omawianego komputera jest mniejsza od prostego zestawu XT, co podwyższa jego atrakcyjność. Gdyby wersja sprzedawana w Polsce miała wbudowany Centronics, pełny RS 232 C oraz gdyby istniał zaimplementowany dla SAMa CP/M (najlepiej wersja 3.0 Plus), to komputer ten miałby przed sobą jeszcze większą przyszłość. Jednak po dokupieniu odpowiedniego interface'u może on być wykorzystywany do wielu zadań, takich jak edycja tekstów, czy nawet edukacja. O ile cena SAMa COUPÉ nie wzrośnie — zakup jego wart jest przemyślenia.

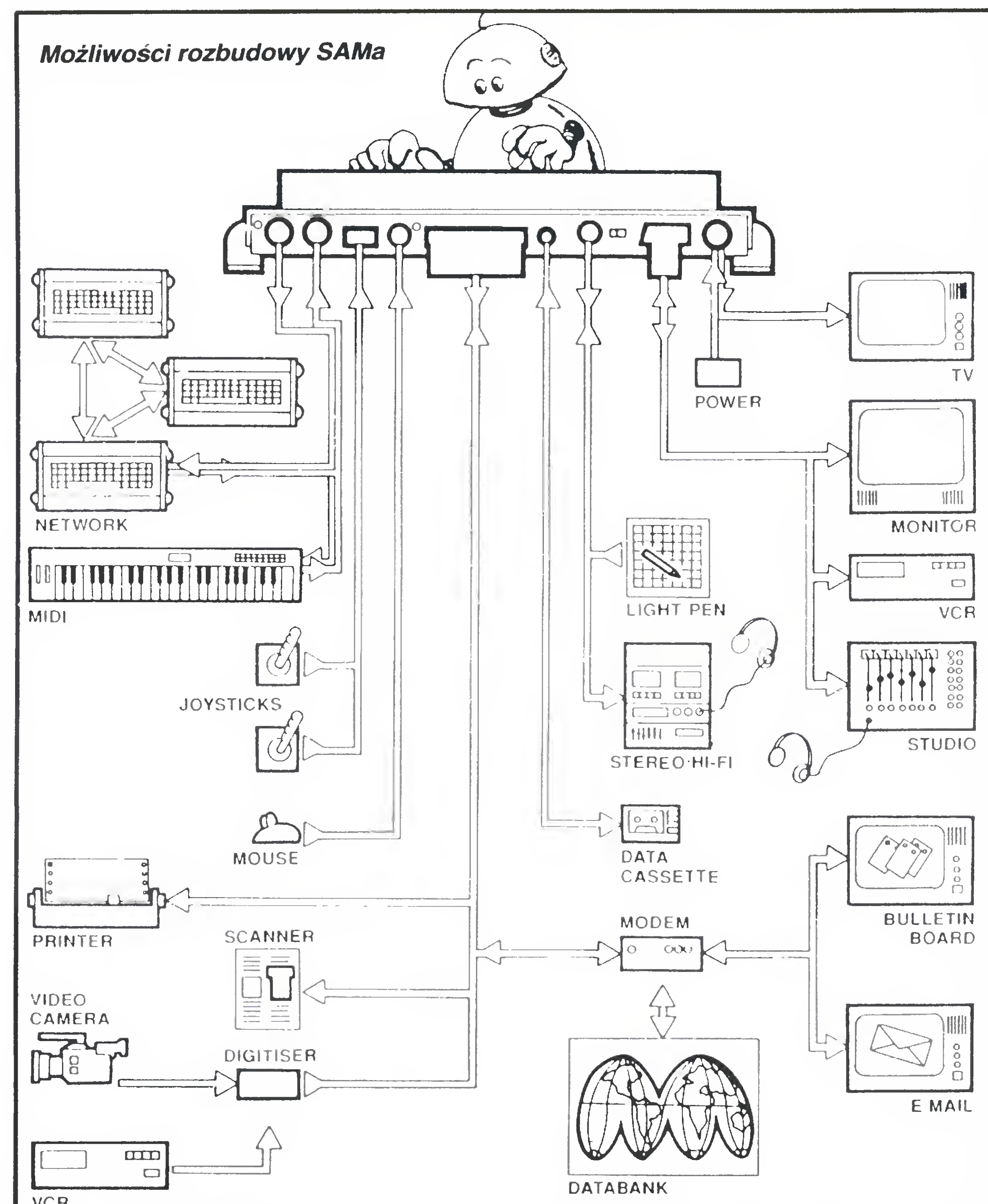
Maciej Pietraś

Ocena:

Łatwość posługiwania się	**3/5**
Dokumentacja	**2/5**
Parametry techniczne	**3/5**
Szybkość działania	**3/5**
razem	**11/20**

Tabela danych technicznych SAMa COUPÉ:

Procesor:	Z80B o szybkości 6MHz
Pamięć RAM:	256KB (rozbudowa do 4.5MB)
Pamięć ROM:	32KB zawierające SAM BASIC i BIOS
Dźwięk:	układ dźwiękowy Philips SAA 1099; 6 kanałów, osiem oktaw stereo, regulowane obwiednie i głośność
Grafika:	Motorola MC 1377P Video Chip, cztery tryby graficzne: [1] identyczny jak ekran ZX Spectrum [2] jak tryb [1], lecz zwiększone możliwości atrybutów koloru [3] 512 x 192 punktów ekranu graficznego, każdy punkt w innym kolorze, lecz w linii tylko 4 kolory z palety 128 [4] 256 x 192 punktów ekranu graficznego, każdy punkt w innym kolorze, lecz w linii tylko 16 kolorów z palety 128
Wyjścia:	MIDI IN, MIDI OUT, dla myszy, złącze SCART, wyjście joysticka, dla pióra świetlnego, wyjście do rozbudowy systemu.
Dyskietki:	3.5 cala, specjalne napędy firmy Citizen pojemność sformatowanej dyskietki: 780KB



Zestaw SAM COUPÉ otrzymaliśmy od wyłącznego dystrybutora na Polskę „Electronics Export" Londyn tel. (44) 81991 0928, fax: (44) 81998 5914. Komputer ten jest rozprowadzany w Polsce poprzez PHP UNICOMP Sp. Z.o.o. 05-870 Błonie ul. Przybysza 20, tel. (0-22) 554-554 w godz. 10—16 lub (0-22) 257-694 (24 godz.), fax: 254-879, tlx: 813276. Sprzedaż detaliczna: sklepy „Europa" Warszawa oraz wiele sklepów komputerowych na terenie kraju — informacje „UNICOMP".

UWAGA — KONKURS!

i TY możesz mieć SAMa

Główną nagrodą w tym konkursie jest komputer SAM COUPÉ, ponadto można wygrać jeszcze pięć joysticków, lecz należy poprawnie odpowiedzieć na trzy pytania. Nagrody ufundowała firma ELECTRONICS EXPORT z Londynu — wyłączny dystrybutor SAMa na Polskę.

Odpowiadając na pytania prosimy napisać numer pytania i obok odpowiedź. Karty pocztowe należy wysyłać na adres naszej redakcji z dopiskiem „Konkurs BRØMBY". A oto pytania:

[1] Jaki adres w ZX Spectrum ma zmienna LAST K?
[2] Jak się nazywa program graficzny dołączany do SAMa?
[3] Czy można podłączyć drukarkę bezpośrednio do SAMa?

BRØMBA

Na odpowiedzi czekamy do 30.09.91

JĘZYK MASZYNOWY cz. 11

W poprzednim odcinku opisaliśmy zmienne systemowe wykorzystywane przez dwie procedury, których omówieniem zajmiemy się dzisiaj.

Przedstawiony obok listing zawiera wydruk procedury **MAKE ROOM** wraz z podprocedurą **TEST ROOM**, procedury **RECLAIMING** łącznie z podprocedurą **DIFFER** oraz wspólnej **POINTERS**. Celem procedury **MAKE ROOM** jest przesunięcie „w przód" określonej części programu w BASIC'u wraz z wykorzystywanym przez system obszarem znajdującym się za programem. Procedura ta aktualizuje związane z przesunięciem zmienne systemowe. Istotą algorytmu jest skontrolowanie czy istnieje dostateczna ilość wolnej pamięci zapewniająca prawidłowe wykonanie przesunięcia. Jeśli istnieje potrzebna ilość wolnych bajtów, to procedura aktualizuje właściwe zmienne systemowe a następnie przepisuje bajty przesuwanej części programu i obszarów znajdujących się za programem w „nowe" miejsce oddalone od „starego" o wartość z rejestru **BC**.

Gdy chcemy wywołać procedurę musimy wpisać do rejestru **BC** ilość bajtów żądanej wolnej przestrzeni, w rejestrze **HL** adres początku tej przestrzeni. Główna procedura składa się z czterech części: część pierwsza (rozkazy 1—3) nie wymagają wyjaśnienia, część druga (rozkazy 4—15) testuje wolną pamięć. Częścią trzecią jest podprocedura modyfikująca adresy zmiennych systemowych dotyczących „przesuwanej" części programu. Nosi ona nazwę **POINTERS** i dokonuje przeglądu czternastu dwubajtowych zmiennych (od 23627 do 23653). Modyfikowane są tylko te zmienne, w których znajdujące się adresy są równe lub większe od adresu początku (będącego w HL). **POINTERS** jest ciekawą podprocedurą i możemy ją opisać dokładniej. Podprocedura na początku pobiera adres pierwszej z czternastu zmiennych a jej zawartość ładowana jest do **DE**. Rozkaz 21 przygotowuje licznik przeglądanych zmiennych. Dalej podprocedura aktualizuje adresy w wybranych zmiennych systemowych (rozkazy 30—42). Warunkiem wejścia do tej części podprocedury jest spełnienie warunku, aby adres zawarty w danej zmiennej dotyczył obszarów przesuwanych (HL=<DE). Inaczej nie zostanie ustawiony wskaźnik CY, wykonywany jest rozkaz 30 i część rozkazów zostaje pominięta. Aktualizacja adresu w danej zmiennej polega na dodaniu do niego wartości zawartej w **BC**.

Następnie przygotowywana jest pełna ilość bajtów do przesunięcia. W tym celu do **DE** pobiera się adres początku wolnej przestrzeni, odłożony na stosie rozkazem 19 (w HL jest „stary" adres wolnej przestrzeni). Rozkazy 47, 48 i 49 ustalają ilość bajtów pamięci podlegających przesunięciu. Aby rozkaz przesyłania grupowego (LDDR/LDIR/) działał poprawnie należy rejestr **BC** zwiększyć o jeden. Po inkrementacji **BC** i przywróceniu w **HL** „starego" adresu wolnej przestrzeni — rozkaz 51 — podprocedura **POINTERS** kończy swe działanie i następuje powrót do procedury głównej, gdzie realizowana jest jej czwarta część.

Przesunięcie określonej w rejestrze **BC** ilości bajtów w kierunku wzrastających adresów jest realizowane przez pobranie do pary rejestrów **HL** „nowego" adresu wolnej przestrzeni pamięci, będącego adresem docelowym dokąd bajty zostaną przesunięte. Adres skąd należy je pobrać znajduje się w rejestrze **DE** i został dostarczony przez podprocedurę **POINTERS**. Samo przesłanie bajtów realizuje rozkaz **LDDR**, więc powyższe adresy muszą zostać zamienione miejscami — rozkaz 55. Po zakończeniu działania rozkazu przesłania zostanie wolna przestrzeń w programie, co miała właśnie uczynić procedura **MAKE ROOM**.

Procedura **RECLAIMING** jest alternatywną w stosunku do procedury MAKE ROOM, ponieważ wykonuje odwrotną czynność. Skraca ona program w BASIC'u o określoną ilość bajtów.

Na podstawie adresu początku (np. likwidowanej linii) podanego w **DE** i adresu końca podanego w **HL** procedura ustala długość obszaru programu do „zlikwidowania". Operację zmniejszenia adresów w odpowiednich zmiennych powinna wykonać podprocedura **POINTERS**, dlatego też wartość w **BC** musi być ujemna. W tym celu wartość w **BC** należy zapisać w konwencji zapisu liczb ujemnych czyli kodzie uzupełnień do dwóch. Następnie trzeba zaktualizować zmienne i przepisać odpowiednią część pamięci tak, aby zlikwidować usuwany obszar programu.

Podobnie jak poprzednią, tą procedurę też można podzielić na cztery części. W części pierwszej procedura ustala ilość bajtów do usunięcia. Realizowane jest to przez podprocedurę **DIFFERENCE** (etykieta DIFFER) o adresie 6621 (hex 19DD). Rozkazy jej oznaczono numerami 2–8. Po ustaleniu w **HL** ilości bajtów do przesunięcia i przepisaniu jej do **BC** zostaje odtworzony adres wejściowy w **HL**.

Część druga głównej procedury zamienia dodatnią liczbę zawartą w **BC** na ujemną. Realizują to rozkazy 9–16. Zamianę danej liczby na jej uzupełnienie do dwóch wykonuje się przez negację wszystkich bitów i dodanie jedności.

Część trzecia: aktualizacja zmiennych systemowych przy użyciu podprocedury **POINTERS**. Działanie tej podprocedury jest identyczne jak dla procedury **MAKE ROOM**, jednak rozkaz 33 realizując dodawanie liczb z przeciwnym znakiem w wyniku daje różnicę. Po wyjściu z **POINTERS** w **HL** jest adres początku likwidowanej przestrzeni. Część czwarta procedury wiodącej powoduje przepisanie („dosunięcie") części programu za „zlikwidowaną" przestrzenią do jego części początkowej. Realizują to rozkazy 54–59. Rozkazy 54, 55 i 56 ustalają adres początku likwidowanej przestrzeni (w DE) oraz adres końca (po wykonaniu dodawania jest on w HL) przez dodanie do adresu początku „bezwzględnej" długości (przed dodawaniem jest ona w HL) likwidowanej przestrzeni. Wartość ta została ulokowana na stosie rozkazem 9. Po odesłaniu na stos adresu początku (do przechowania) rozkaz 58 realizuje przepisywanie, lecz zauważmy, że od adresu początku („pod ten adres") wpisywane są bajty od adresu końca („spod tego adresu") likwidowanej przestrzeni. Licznikiem jest wartość BC, która została ustalona rozkazem 47. Jest to więc wartość łączna tzn. likwidowanej przestrzeni, końcowej części programu znajdującej się za „likwidowaną" przestrzenią i obszarów zza programu. Ponieważ przepisywane są bajty zza likwidowanej przestrzeni — stąd po przepisaniu wszystkich, przepisywanie trwa dalej z tym, że w miejsce powtórzonych końcowych bajtów obszaru zza programu wpisywane są zera. W taki sposób została stworzona nowa, dodatkowa część faktycznie wolnej przestrzeni pamięci tzn. zawierająca zera.

W taki oto sposób omówiliśmy mało interesujące, lecz ciekawe z programistycznego punktu widzenia procedury. Następne odcinki niniejszego cyklu będą już mniej „beletrystyczne" i zawierać będą schematyczne opisy przydatnych procedur z ROM'u spectrumowskiego.

Piotr Sumara

```
MAKE-ROOM
---------------------
01 OS LD   BC,0001
02 MR PUSH HL
03    CALL 04,TR
---------------------
04 TR LD   HL,(23653)
05    ADD  HL,BC
06    JR   C,14,R4
07    EX   DE,HL
08    LD   HL,0000
09    ADD  HL,DE
10    JR   C,14,R4
11    SBC  HL,SP
12    RET  C
---------------------
13 R4 "Out of memory"
14 R4 LD   L,03
15    JP   0055,ERR-3
16    POP  HL
17    CALL 18,PS
---------------------
54    LD   HL,(23653)
55    EX   DE,HL
56    LDDR
57    RET
---------------------
Znaczenie etykiet
=================

OS-ONE-SPACE

MR-MAKE-ROOM

TR-TEST-ROOM

R4-REPORT-4

PS-POINTERS
```

```
POINTERS
---------------------
18 PS PUSH AF
19    PUSH HL
20    LD   HL,23627
21    LD   A,14
22 PN LD   C,(HL)
23    INC  HL
24    LD   D,(HL)
25    EX   (SP),HL
26    AND  A
27    SBC  HL,DE
28    ADD  HL,DE
29    EX   (SP),HL
30    JR   NC,38,PD
31    PUSH DE
32    EX   DE,HL
33    ADD  HL,BC
34    EX   DE,HL
35    LD   (HL),D
36    DEC  HL
37    LD   (HL),E
38 PD INC  HL
39    POP  DE
40    INC  HL
41    DEC  A
42    JR   NZ,22,PN
43    EX   DE,HL
44    POP  DE
45    POP  AF
46    AND  A
47    SBC  HL,DE
48    LD   B,H
49    LD   C,L
50    INC  BC
51    ADD  HL,DE
52    EX   DE,HL
53    RET
---------------------
```

```
RECLAIMING
---------------------
01 R1 CALL 02,DR
---------------------
02 DR AND  A
03    SBC  HL,DE
04    LD   B,H
05    LD   C,L
06    ADD  HL,DE
07    EX   DE,HL
08    RET
---------------------
09 R2 PUSH BC
10    LD   A,B
11    CPL
12    LD   B,A
13    LD   A,C
14    CPL
15    LD   C,A
16    INC  BC
17    CALL 18,PS
---------------------
54    EX   DE,HL
55    POP  HL
56    ADD  HL,DE
57    PUSH DE
58    LDIR
59    POP  HL
60    RET
---------------------
R1-RECLAIM-1

DR-DIFFER

R2-RECLAIM-2

PN-POINTERS-NEXT

PD-POINTERS-DONE
```

HAVE A FUN!



SWORD OF SODAN

Electronic Arts

Jeśli interesuje Cię los zrzuconego z tronu dobrego króla, jeśli chcesz zabić złego czarownika Zorka i jeśli do tego chcesz także dobić ledwo żyjący już joystick, to masz ku temu niepowtarzalną okazję. Sword of Sodan to siedmiolevelówka, poziomo scrollowana gra typu „zabij na śmierć". Grafika i muzyka, choć ocenione na zachodzie nadzwyczaj wysoko, nie odbiegają od średniego poziomu. Najważniejsza jest tu zabawa i myślący pierwszy traci głowę — tak na ekranie jak i w rzeczywistości.



CASTLES

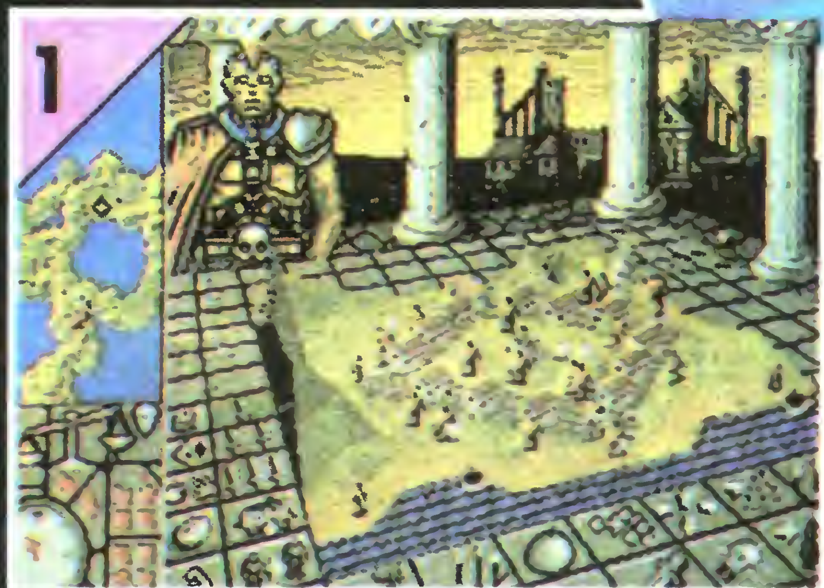
Interplay

Wspaniałe czasy dzieciństwa, gdy bawiłeś się w Indian i kryłeś w gałęziach jak Robin Hood, staną się Twoim udziałem na komputerze dzięki firmie interplay. Sprawdzisz się jako budowniczy wielu twierdz, gdzie animacja i sam sposób tworzenia daje *total* autentyczność. Czasem utrudnią Ci życie czarownice, gotujące w swych kotłach kwaśne zupki — Twe hordy nie zawsze rozwiążą problemy. Nie sposób wyliczyć wszystkich zalet programu, w którym nawet sekwencje budowania zamków są zgodne z rzeczywistością.

ORIENTAL GAMES

Microstyle

Kendo, Kyo Kushinkai i Kung Fu — któż nie zna tych nazw. Typowe, chińskie sztuki walki przeniosła tym razem na ekrany komputerów firma Microstyle, osadzając ją z miejsca w ramach XX-go wieku. Spotkania odbywają się na nowoczesnych arenach i mają raczej formy sparingu, niż walki na śmierć i życie. Poza zwykłymi potyczkami, mamy jeszcze prawdziwy turniej z udziałem mistrzów każdej dyscypliny. Ale szczegóły pozostawmy chłopcom z Top Secret.





POWERMONGER

Bullfrog

Nie minął jeszcze wiatr wywołany przez Populous a na niebie prawdziwa chmura gradowa. Gdy więc udało Ci się pokonać sąsiada w Populous, znaczy, że nadajesz się na Powermongera. Ale uważaj! Do podbicia jest cały świat, a właściwie 195 sektorów mapy i zanim nie zdobędziesz prawego górnego rogu mapy, nie staniesz się Powermongerem. Podporządkowując sobie kolejne wioski, prowadząc wojny z innymi, rzucając swe siły tylko na słabszych masz szanse na zwycięstwo.

LIFE & DEATH II: THE BRAIN

The Software Toolworks

Fani operacji i leczenia, ale także miłośnicy rajcowania na stole operacyjnym z bezbronnymi pacjentami będą usatysfakcjonowani wiadomością o pojawieniu się drugiej części programu Life & Death (tytuł przez sadystów tłumaczony: Żyć aby Umrzeć). Zgodnie z zapowiedziami, gra nie jest skomplikowana ale i nie uproszczona. Osiągnięto to dzięki lepszemu rozplanowaniu przedmiotów i dokładnych instrukcji w oryginalnych (!) wersjach gry. A największym sadystom życzę, by kiedyś naprawdę znaleźli się w szpitalu. Nawiasem mówiąc, stan ich umysłu nie wyklucza tej perspektywy.

● — bronie na gady
1,2,3,4,5 — części talizmanu
V — zgniatacze
⧺ — kraty
□ □ ¿ — hak na linę
⦚ — lina
X — przegroda

A — powietrze
E — energia
G — gad
C — świeca
R — lina
K — klucz
W — czarownik
D — koszulka
M — mapa



GOLDEN TALISMAN

Gry labiryntowo-zręcznościowe nie są popularne i składa się na to kilka przyczyn. Po pierwsze są trudne, co wykrusza mniejszych zapaleńców. Po wtóre nie wyróżniają się grafiką i dźwiękiem, które ogranicza szkielet gry. Po trzecie wreszcie, ukończenie labiryntówki wiąże się z kilogramami straconego czasu.

Golden Talisman to chyba najbardziej typowy przykład takiego programu. Wyprodukowany tak dawno, że prawie archaiczny, może z powodzeniem zniechęcić już po pierwszym pstryknięciu fire. Chyba tylko siły wyższe skłoniły pewne osoby do narysowania jakże trudnej mapy. Mamy nadzieję, że spróbujecie wykorzystać nasz twór; choć może zdobycie gry okaże się trudniejsze od jej skończenia?

Cel gry jest prosty — odnaleźć pięć części Magicznego Talizmanu Foghra, rozrzuconego po podziemnych tunelach. Jeśli tylko nie przerazi Cię od razu napis: „Porzućcie wszelką nadzieję, Wy, którzy tam wchodzicie".

Cóż oprócz mapy może być Wam przydatne?

— Stwory grasujące po labiryncie, podzielić można na dwie grupy: gady, czyli smoki, oraz płazy i ryby. Smoki blokują najczęściej drogę do kluczy, więc używając wcześniej znalezionych broni, takich jak np. krzyż, niszcz je. Płazy i cała reszta ma za zadanie pozbawić Cię energii — jedyną Twą ripostą powinien być celny, a co ważniejsze szybki strzał.

— Na najniższych poziomach znajdują się siły nieczyste, zajmujące się pilnowaniem pustki. Zabicie ich możliwe jest tylko przy użyciu świecy oraz nieodzownej w pewnych przypadkach magicznej koszulki.

— Dopuszczono tylko pięcioprzedmiotową kieszeń, w której używany jest przedmiot z prawej strony.

— Owoce dostrzeżone po drodze przyczynią się do odzyskania sił, niebieskie kubełki natomiast zwiększą zapasy tlenu niezbędnego pod wodą.

— Warto zbierać klucze, którymi otworzysz kraty, oraz liny, które zaczepisz w zaznaczonych sklepieniach. Mapy natomiast przydadzą się do rozwarcia kamiennych przegród.

— Świece oświetlają ciemne pomieszczenie.

— Wyjście z wody jest możliwe tylko po linie.

— Upadek z dużej wysokości, podobnie jak nagłe zetknięcie z wodą, nie są dla Ciebie groźne.

— Smoki na najniższym poziomie można przejść dzięki przestrzeni pod ich skrzydłem. Po zabiciu jeden z takich gadów zamieni się w świecę.

— Jedno trafienie czarownika zabiera 10% energii, dotknięcie potworka zaś 1/10%.

— Pamiętaj, że tylko przy pomocy Magicznego Talizmanu Foghra zabijesz ponurego Pana Labiryntu, kończąc tym samym pomyślnie całą misję.

— Często jeden dokładny lub niedokładny skok może zadecydować o losach wielogodzinnej gry. Bądź więc uważny — bo to nie są czcze przestrogi. Powodzenia!

Luke

Firma: nieznana
Rok produkcji: 1986
Komputer: Spectrum, Commodore, Atari

DEJA VU II

(Już to gdzieś, kiedyś... No, głowę bym dał!)

Oto obraz faceta grającego w tekstówkę: siedzi przygarbiony nad klawiaturą i klepie. Czasem chwilę się zastanawia, potem znowu klepie, co rusz potykając się o komunikaty w rodzaju „I don't understand word stupid", „that idea doesn't work" i tym podobne. Smutny to widok.

Ale na szczęście coraz rzadszy. Wielka to zasługa takich programów jak Deja Vu i zanim zajmiemy się fabułą gry, zobaczmy, jakie reguły rządzą jej obsługą.

Przede wszystkim słownik, który autorzy ograniczyli do ośmiu poleceń: EXAMINE (zbadaj), OPEN (otwórz), CLOSE (zamknij), SPEAK (mów), GO (idź), HIT (uderz), CONSUME (spróbuj), i wreszcie najbardziej użyteczna opcja, umożliwiająca wykonywanie wszelkich działań zgodnych z logiką — OPERATE. W innych programach tekstowych odpowiednik jej to USE.

Ciekawa jest organizacja ekranu. Został on podzielony na okna: tekstu, grafiki, rozkazów i przedmiotów (INVENTORY). Grafika w Deja Vu II jest bardzo staranna oraz ważna ze względu na to, że każdy przedmiot bierze na swój sposób udział w grze — można na nim wypróbować działanie wszystkich dostępnych rozkazów i innych przedmiotów. Okna można przesuwać po całym ekranie i podporządkowywać wedle upodobania, zaś przedmioty w oknie INVENTORY są pokazywane graficznie, co działa pobudzająco na ośrodek kojarzeniowy. Mamy ponadto opcję CLEAN UP, która sama uporządkuje rozbabrany inwentarz.

Jak nietrudno się domyśleć, cała gra obsługiwana jest myszą, a to stanowi znaczący krok wstecz. Dlaczego? To przecież jasne — siedząc o krok dalej od monitora oszczędzamy oczy.

Akcja Deja Vu II przenosi nas w lata pięćdziesiąte, wcielając w postać prywatnego detektywa drugiej kategorii — Ace'a Hardinga. Został on właśnie porwany z rodzinnego Chicago do Las Vegas, gdzie mafioso Tony Malone po przyjacielsku wyłożył mu swoje kłopoty. Otóż w Chicago zginął niejaki Joey Siegel, który zbierał tam forsę dla Malone'a i przesyłał ją kurierem do Las Vegas. Starego Siegela można odżałować, ale nie przeszło sto tysięcy zielonych, które wsiąkły gdzieś w zamieszaniu. A Harding miał nieszczęście kręcić się zbyt blisko całej sprawy, więc teraz troski Malone'a stały się także troskami Ace'a. Albo dostarczy brakującą sumę Tony'emu w ciągu tygodnia, albo...

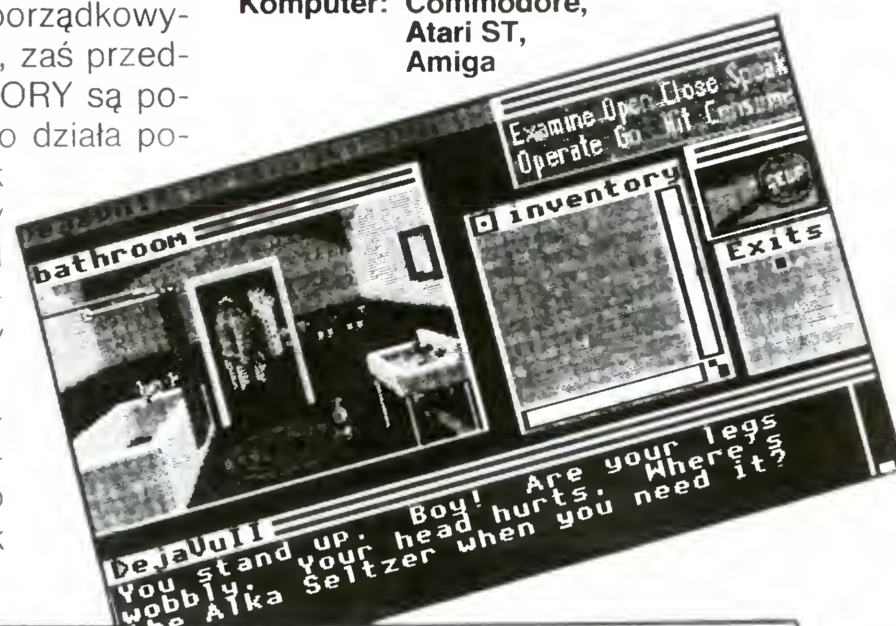
Twoja w tym głowa, żeby w odpowiednim czasie pieniądze znalazły się we właściwych rękach.

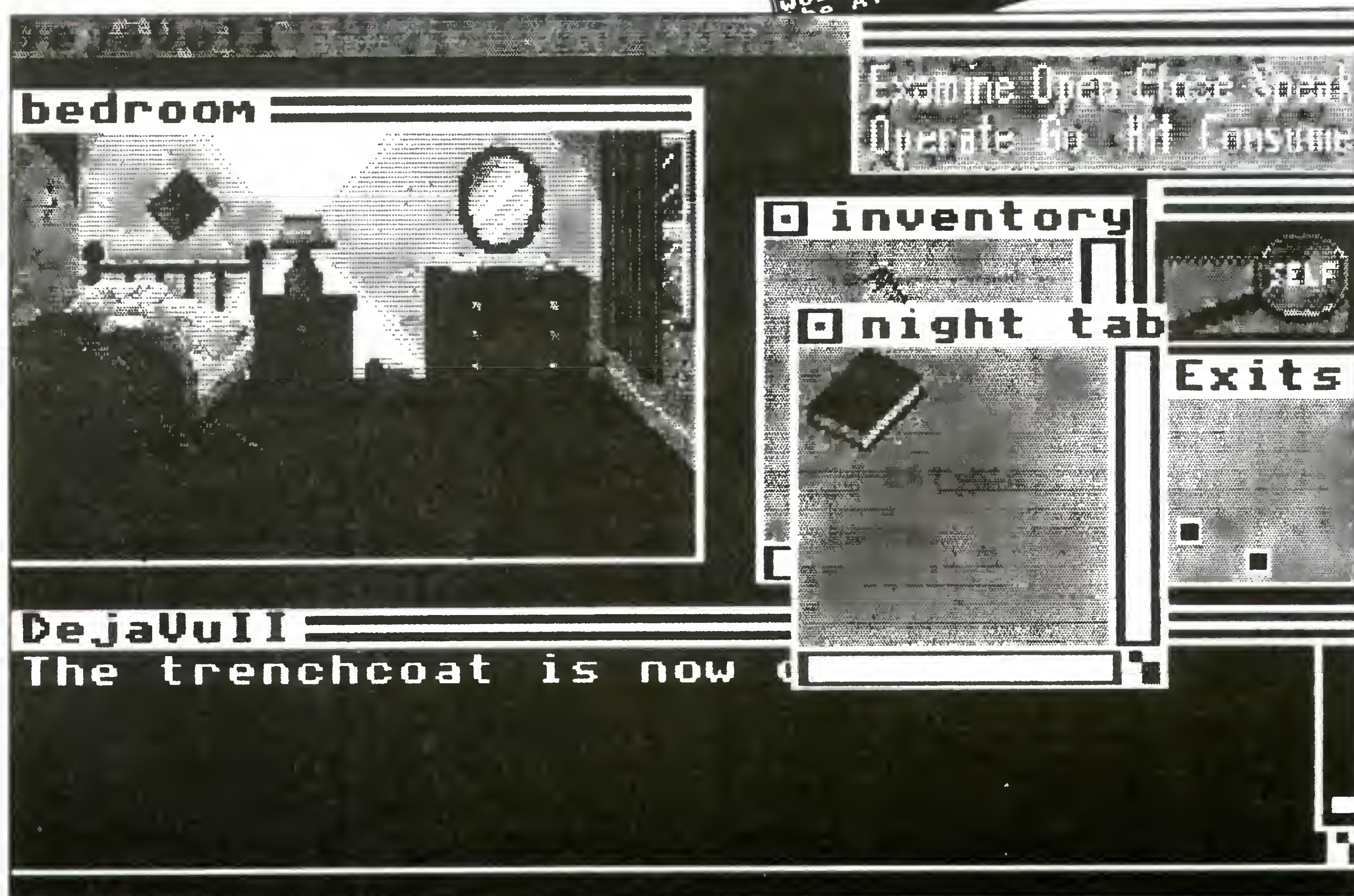
Gra ma typowo Chandlerowski nastrój, tak jeśli chodzi o jej akcję i rodzaje kruczków, tak jak i grypsy rzucane przez komputer. Uśmiech ekstazy wywołują takie komentarze komputera: „zastanawiam się, czy te kraty służą do trzymania kasjerki wewnątrz, czy takich ludzi jak ty na zewnątrz", lub „zlew jest brudny i śmierdzi stęchlizną; przypomina ci o domu".

I to już chyba wszystko.

John Leon

Rok produkcji: 1990
Komputer: Commodore, Atari ST, Amiga





KOMPUTER	Giełda	Sklep	Pewex/Baltona
	tys. zł	tys. zł	tys. zł
SINCLAIR			
ZX Spectrum 48	1100	—	—
ZX Spectrum +	1200	—	—
ZX Spectrum +2	—	—	—
ZX Spectrum +3	2500	—	—
Timex 2048	1250	—	—
FDD 3000	1250	—	—
Sam Coupe	—	3499	—
COMMODORE			
C 64	1310-1410	1700	1750
C 128	2200	—	—
Amiga 500	4350-4500	5999	—
Amiga 2000 C	9200	—	—
Amiga 3000	39000-44000	—	—
1 MB do Amigi	500-640	699	—
Magnetofon	200-240	399	390
1541-II	1650	1999	—
1571	1700 (używ.)	—	2300
Monitor 1084S	3400	—	4390
Monitor 1802	2300	4000	—
Stacja 3.5" Amiga	900	1249	—
Stacja 5.25" Amiga	1200	2079	—
Modulator TV	280	500	—
PC 20 III	—	—	11890
PC 10 III	—	—	8890
HD 20 do Amigi	—	—	—
ATARI			
800 XL/XE	1200 (XE)	—	—
65 XE	1300	1999 (+magn.)	1590
130 XE	1500	2200	2390
520 STFM	3999	4999	—
520 STE	5400 (1 MB)	6449	7690
1040 STFM	4250	5199	8890
Mega 2	13000 (STE)	—	16890
Mega 4	—	—	24390
CA 2001	1900	—	2490
SM 124	1600	2850	—
SM 224	3700	—	4690
Magnetofon	290	399	510
Portfolio	2300	2999	5890
AMSTRAD			
CPC 464	1400	—	—
CPC 664	—	—	—
CPC 6128 mono	2800	—	—
CPC 6128 color	3500	—	—
PCW 8256	—	—	—
IBM			
XT	2800-5500	4000-6000	5000
AT	5000-12000	4700-11800	8990
386	17000	12900-19500	—
486	—	25200-38200	—
Laptop	—	—	23790
HD 40	2500-3500	2800-3000	4990
Monitor CGA mono	1000 (używ.)	—	—
Monitor VGA mono	1280	—	—
Monitor VGA color	3300-4300	—	—
Klawiatura	450	660	—
INNE			
Dyski 3" (szt.)	35	60	—
Dyski 3.5" (szt.)	6.5-19	8-39 (HD)	20
Dyski 5.25" (szt.)	3.6-16	5.5-25 (HD)	15
Joystick	46-416	100-450	99
Pudełko 100 3.5"	85	100-150	—
Pudełko 100 5.25"	85	100-150	—
Monitor Philips	3300	—	—
DRUKARKI			
Star LC 20	2400	2500	—
Star LC 200	3400 col	3500	—
Star LC 24-200	4600-5100 col	—	—
Commodore MPS 1270	1750	—	—

Sklepy „Bajtka": Bytom, ul. Kolejowa 6, tel. (832) 81-49-17, Kraków, ul. Kalwaryska 9, tel. (012) 22-59-72

Od pewnego czasu giełda warszawska przeniosła się na trzeci róg kwadratu, którego pierwszymi dwoma były stara giełda w szkole podstawowej i giełda w domu kultury. Z zewnątrz nowa szkoła podstawowa przypomina trochę tę pierwszą, ale po wejściu do środka podobieństwo jest prawie stuprocentowe — brak tylko dużej sali gimnastycznej oddanej do dyspozycji sprzedających. Mimo tego sprzedających jest mniej, może dlatego, że dom kultury był dość mały, mniejszy od obecnej lokacji, a nie wszyscy zdołali się przenieść — drugie piętro jest prawie całe puste. Handel sprzętem pozostał jednak po drugiej stronie ulicy Jana Pawła II i tu wielkie brawa dla organizatorów giełdy — teren ogrodzono i ul. Grzybowska nie będzie już co tydzień zaśmiecana. Kupujących jak i sprzedających jest mniej niż normalnie — no cóż, trwają wakacje...

Komentarz ten pisany jest bezpośrednio przed pierwszą giełdą sierpnia, czyli już po wprowadzeniu nowej taryfy celnej. Ceny sprzętu powinny trochę się podnieść i szczęśliwi są ci, którzy zakupy zrobili wcześniej. Mimo to dalej niesłabnącym zainteresowaniem cieszy się Amiga i wszelki osprzęt jak i oprogramowanie do tego komputera, którego jest pełno na każdym kroku. Inne komputery też są popularne, należy tu zwrócić uwagę na swoisty fenomen IBM PC gdyż ostatnio przestał być komputerem tylko do pracy i coraz częściej można na nim zagrać. No cóż, nawet Big Blue nie przeciwstawił się szponom hazardu...

W następnej giełdzie przedstawimy aktualne ceny już po wejściu nowych ceł w życie. Dane zebrano dnia 20.07.91 r.

Drodzy Czytelnicy!

Jak zapewne zauważyliście, zmieniła się nieco formuła indywidualnego Banku Danych. Napływa do nas bardzo wiele listów z prośbą o publikację w tej rubryce; z konieczności więc musimy skracać treść ogłoszeń, co pozwoli nam spełnić prośby większej liczby Czytelników i tym samym zlikwidować zaległości czasowe tej rubryki.

Zmianie ulegną także rubryki SOS oraz KUPIĘ-SPRZEDAM-ZAMIENIĘ. Zamiast szeregu oddzielnych kuponów od numeru 06/91 wprowadzamy jeden kupon ważny dla wszystkich rubryk; jego nadesłanie w liście uprawnia do publikacji krótkiego ogłoszenia (o pierwszeństwie decyduje data stempla pocztowego) w jednej z tych rubryk. Nadesłanie większej liczby kuponów uprawnia do opublikowania ogłoszenia w kilku rubrykach, również w zależności od daty nadania listu.

Ze względów technicznych redakcja zastrzega sobie prawo skracania ogłoszeń do niezbędnego minimum. Podczas pisania kierujcie się nowym formatem IBD; chcielibyśmy, aby w tych rubrykach ukazywała się większa liczba ogłoszeń.

I jeszcze jedno. Piszcie na kopertach, do jakiej rubryki jest skierowany list oraz (także na kopercie, np. w lewym dolnym rogu) typ komputera. To bardzo ułatwi sprawę naszej Pani Sekretarce.

IBM

Marek Futrega (13), PC/XT. Proponuje wymianę oprogramowania, literatury i doświadczeń. **A**: Wróblewskiego 12/3, 14-300 Morąg, woj. olsztyńskie.

—

Grzegorz Piekarz, PC/AT. Proponuje wymianę oprogramowania. **A**: B. Głowackiego 25/11, 85-614 Bydgoszcz.

—

Marcin Drozdek, Amstrad PC 640 DD. Proponuje wymianę gier, szuka Populous i King's Quest. **A**: Tomaszewicza 3/59, 94-048 Łódź.

COMMODORE

Wiesław Paciukanis, C-64. Poszukuje literatury ogólnej oraz dotyczącej C-64. **A**: Kowalskiego 23b/17, 16-400 Suwałki.

—

Aleksander Latusek (21), A500 1MB RAM, LC20. **P**: około 1000 dyskietek. Proponuje wymianę **P. A**: Zwrotna 14, 40-610 Katowice.

—

Dominik Kobus (16), A500. Proponuje wymianę **P**, **L** i **D**. **A**: Prądzyńskiego 19/2, 07-400 Ostrołęka. **T** 47-38.

—

Grzegorz Płaczek (13), C-64 + magnetofon. Proponuje wymianę **P** i **D**. **A**: Opolska 28/3, 42-600 Tarnowskie Góry.

—

Grzegorz Komar, C-64 + 1571. Poszukuje członków do swojej grupy. **A**: Karabeli 15/19, 01-313 Warszawa.

ZX SPECTRUM

Frantisek Tunega (46), Spectrum. **P**: 500 gier w tym 25 szachowych. Proponuje wymianę oprogramowania. **A**: SNP 774, 53803 Hermanuv Mesteo, CSFR.

ATARI

Wojtek Piekutowski, 65XE + XC-12 (Turbo 2000). Proponuje wymianę **P** i **D**. **A**: Gwardii 8a, 21-300 Radzyń Podl. **T**: 520-067.

KLUBY KOMPUTEROWE

KLUB AMIGA KANGU — wymiana doświadczeń — programy narzędziowe. **A**: Andrzej Sicinski, Platterstrasse 52, 6200 Wiesbaden, Deutschland.

—

KLUB „ELEKTRONIK" — nawiąże kontakty z posiadaczami ZX Spectrum. **A**: Klub „Elektronik", A.B. Makarow, Odessa — 114 a/292, 270114 ZSRR.

—

Grzegorz Płaczek (13), zamierza założyć klub komputerowy. **A**: Opolska 23/3, 42-600 Tarnowskie Góry.

INNE KOMPUTERY

Jan Krawczyk, MK-45. Nawiąże kontakt z użytkownikami tego komputera. **A**: Spółdzielcza 1/15, 21-040 Świdnik. **T**: 124-42.

KUPIĘ ● SPRZEDAM ZAMIENIĘ

Amiga

1. Kupię Amigę 500 w rozsądnej cenie. R. Bykowski, ul. Wewnętrzna 6, 05-410 Józefów, tel. 793011 w 123.
2. Kupię oprogramowanie na Amigę 500. M. Gruszecki, os.Łużyckie 2A/7, 68-200 Świebodzin.
3. Sprzedam Amigę 1000, mysz, joystick, kartę RAM 4MB, literaturę (4 mln.). P. Krzemiński, ul. Gwardii Ludowej 4/10, 27-400 Ostrowiec Św, tel. 26265.

Amstrad

1. Kupię stację dysków do Amstrada CPC 464. R. Marciniak ul. Kosmonautów Polskich 23/8, 67-200 Głogów tel. 333476 w 375.
2. Kupię, sprzedam, zamienię programy na CPC (dyskietka lub taśma). Kupię mysz i modulator. G. Teresiński, Al. Racławickie 4/7, 20-028 Lublin, tel. 28949.
3. Odstąpię instrukcję obsługi komputera Amstrad CPC 6128 (w j. polskim). B. Myszkiewicz, Warszawa, tel. 177210.
4. Sprzedam Amstrada CPC 464 z kolorowym monitorem. K. Wróblewska, ul. Przelot 60, 60-408 Poznań.
5. Sprzedam Amstrada CPC 464, zielony monitor i instrukcję. W. Mazgis, ul. Inżynierska 69/12, 53-230 Wrocław.
6. Sprzedam CPC 464, 2 stacje 3", monitor kolorowy, oryginalny wzmacniacz+głośniki, 30 dyskietek i literaturę (komplet 5.6 mln). B. Tryt, ul. Poznańska 46f/2, 66-600 Krosno Odrz.
7. Sprzedam CPC 6128, kolorowy monitor, drukarkę DMP 2000, stację 5.25" (5.5 mln). K. Mazanowski, Wrocław, tel. 252641 w.181 do 16[00], tel. 482915 po 16[00]
8. Sprzedam półrocznego Amstrada CPC 6128 z kolorowym monitorem. M. Patynowski, ul. Abrahama 5/45, Warszawa, tel. 125898

Atari

1. Kupię Atari ST, Amigę, IBM PC lub podobne w dowolnej konfiguracji. B. Sołobodowski, 14-100 Ostróda, skr. poczt. 28.
2. Kupię drukarkę Atari 1029 z polskimi literami. M. Kołodziejski, ul M. C. Skłodowskiej 26, 96-100 Skierniewice.
3. Kupię kabel lub dwie wtyczki łączące Atari XE ze stacją 2001. P. Chrapowicz, ul. Reymonta 2/47, 18-400 Łomża.
4. Kupię książkę W. Zientary "Poradnik programisty Atari XL/XE, Podstawy programowania w Atari Basic. A. Zajączkowski, 77-235 Trzebielino.
5. Sprzedam Atari 130 XE, CA 12, AST, joysticki, literaturę. T. Szadkowski, ul. Zdrojowa 12, 99-300 Kutno, tel. 33454.
6. Sprzedam Atari 65 XE, XCA 12+Blizzard, Turbo+cartridge Phoenix. W. Ścigała, ul. Brożka 13/30, 41-902 Bytom.
7. Sprzedam Atari 65 XE i CA 12. T. Pelc, Wola Dalsza 342, 37-100 Łańcut.
8. Sprzedam Atari 65 XE na gwarancji, CA 12 z Turbo 2000 i cartridge M. Kosiński, ul. Piecewska 18a/2, 80-288 Gdańsk.
9. Sprzedam Atari 65 XE z magnetofonem. A. Kowalczyk, Chełm, tel. 58878.
10. Sprzedam Atari 65 XE z magnetofonem. P. Wydymus, ul. Nowopogońska 53/29, 41-200 Sosnowiec.
11. Sprzedam Atari 65 XE z magnetofonem. Pilne. W. Zawada, ul. Dekutowskiego 12/83, 39-400 Tarnobrzeg.
12. Sprzedam Atari 65 XE, CA 2001 (gwarancja) CA 12 z Turbo, AST cartridge. M. Król ul. Hamernicka 6/22, 26-900 Kozienice.
13. Sprzedam Atari 65 XE, CA 2001, CA 12, monitor (gwarancja). R. Deptuła, ul. Staromłyńska 12, 83-050 Kolbudy, tel. 827267
14. Sprzedam Atari 65 XE, gwarancja, magnetofon. B. Sikora, ul. Drzymały 3/13, Wolsztyn, tel. 444 w 291.
15. Sprzedam Atari 65 XE, LDW 2000 SUPER, XC 12, monitor. P. Burchard, ul. Egejska 5/48, Warszawa, tel. 426761.
16. Sprzedam Atari 65 XE, magnetofon Turbo 2000, monitor, joystick (230$). M. Strawiński, ul. Armii Krajowej 49, 81-870 Sopot, tel. 513695.
17. Sprzedam Atari 65 XE, magnetofon z Turbo AST, cartridge i 2 joysticki. M. Bujnowski, ul. Broniwoja 12/55, 02-655 Warszawa, tel. 437086.
18. Sprzedam Atari 65 XE, magnetofon, Turbo 2000, monitor, 2 joysyicki. M. Zawadzki, ul. Radna 15/8, 00-341 Warszawa, tel. 262274.
19. Sprzedam Atari 65 XE, stację 1050, XC 12 (Turbo), monitor TWM-315. M. Łapka, ul.Brzezińska 116a, 97-140 Koluszki.
20. Sprzedam Atari 65 XE, Turbo CA 12 joystick (2.2 mln). A. Kotkowski, ul. Rojna 50d/60, 91-134 Łódź.
21. Sprzedam Atari 65 XE, XC 12 (Turbo Blizzard). M. Niedworok, ul. Orlików 19, 40-663 Katowice, tel. 528908.
22. Sprzedam Atari 65 XE, XC 12 Turbo 2000. P. Pachołek, ul. Kłodzka 11, 57-521 Gorzanów, tel.92.
23. Sprzedam Atari 65 XE, XC 12 z Turbo 2001 na cartridge'u, joystick (2.2mln.). W. Ruszkiewicz, ul. Kusocińskiego 15/96, 39-300 Mielec.
24. Sprzedam Atari 65 XE, XC 12+Turbo 2000, joystick, literaturę. M. Kusiński, ul. Częstochowska 3/4, 97-570 Przedbórz.
25. Sprzedam Atari 65 XE, XC 12, joystick. M. Jastrzębski, ul. Wiankowa 5, 15-156 Białystok.
26. Sprzedam Atari 65 XE, XC 12, LDW SUPER 2000. J. Wojtas, ul. Żeromskiego 24/13, 64-800 Chodzież, tel. 821221.
27. Sprzedam Atari 65 XE, XC 12, w systemie Turbo 2000. P. Sołtysiak, 42-217 Częstochowa, skr.poczt. 559.
28. Sprzedam Atari 65 XE, XCA 12 w Turbo 2001, literaturę (2.4 mln). R. Fąfara, ul.Przeskok 4/22, 26-600 Radom.
29. Sprzedam Atari 65 XE, XCA 12 z Turbo 2000, joystick. R. Banach, Gdynia tel. 237859.
30. Sprzedam Atari 65 XE, XCA 12, cartridge z Action i joysticki VG 128, SV 125 Megaboard. J. Gomułka, ul. Dekabrystów 12/3, 45-061 Opole.
31. Sprzedam Atari 800 XE, CA 12, Turbo 2001 (gwarancja, instrukcja po polsku). M. Adamczewski, ul. Gałczyńskiego 14/69, 09-400 Płock.
32. Sprzedam Atari 800 XL (256KB RAM), Atari 1050 (HAPPY DRIVE), 2 joysticki, 15 dyskietek. W. Kostka, ul. Golska 41/6, 83-233 Jarczewo
33. Sprzedam Atari 800 XL, 1029, XC 12, CA 2001, dyskietki, joysticki. R. Sakierski, Grudziądz, tel. 21772.
34. Sprzedam Atari 800 XL, magnetofon 1010 (Turbo), joystick. M. Bociański. ul. Polna 40/3, 97-200 Tomaszów Maz. tel. 2581.
35. Sprzedam Atari 800 XL, magnetofon 1010 turbo, 2 cartridge z systemami turbo AST i OS-2, joystick i literaturę. W. Cyrański, os. Chrobrego 21/24, 60-681 Poznań, tel. 223914.
36. Sprzedam Atari 800 XL, XC 12, joystick. A. Grzegorczyk, ul. Piotrkowska 181/5, 90-447 Łódź, tel. 360068.
37. Sprzedam Atari ST 1 MB-komplet. D. Czechowski, ul. K. Wielkiego 15/40, 44-224 Knurów.
38. Sprzedam CA 12 z ASTurbo i multi cartridge AST (gwarancja). P. Styś, ul. Kościuszki 7, 07-140 Sadowne.
39. Sprzedam interface pozwalający na pracę Atari z magnetofonem niefirmowym (80 tys.). K. Łuczak, 98-310 Czarnożyły 11.
40. Sprzedam komplet: Atari 130 XE, XCA 12 z Turbo ROM, joysticki i literaturę (tanio). P. Laszczyk, 34-511 Kościelisko 976, Zakopane.
41. Sprzedam literaturę i oryginalny cartridge z "Donley Kong" do Atari 65,800 XL/XE. H. Tuszyński, ul Polaka 3/117, 02-777 Warszawa, tel. 6413418.
42. Sprzedam lub zamienię na Amigę 500, Atari 800 XL, magnetofon w Turbo AST itd. M. Bogdan, ul. Korczaka 3/63, 16-400 Suwałki.
43. Sprzedam lub zamienię na C 64, Atari 65 XE z XC 12 (Turbo 2000). A. Herbaczyński, ul. Kasprzaka 71/11, 05-200 Wołomin.
44. Sprzedam tanio Atari 65 XE, XC 12 w Turbo 3000, 2 joysticki. P. Pęcherz, ul. Wiejska 8/2, 58-530 Kowary.
45. Sprzedam tanio nową stację do Atari CA 2001, M. Burda, ul. Sternicza 10, 43-300 Bielsko-Biała.
46. Tanio sprzedam Atari 800 XE z wyposażeniem. J. Grobelny, Łask, tel. 5905 (wieczorem).
47. Tanio sprzedam Atari 800 XL, stację 1050+HAPPY WARP, XC 12. T. Strożek, ul. Magistracka 27/26, 01-413 Warszawa, tel. 363078.
48. Tanio sprzedam roczną stację CA 2001 (stan idealny). I. Dziedzic, ul. Pomorska, 9/36, 84-232 Rumia.

Commodore

1. Kupię C 16 z magnetofonem i joystickami. K. Siemaszko ul. Gołdapska 18/72, 19-400 Olecko, tel. 2873, kier. 88781 z W-wy.
2. Kupię książki o C 64 (muzyka, grafika, przerwania). M. Matussek, ul. Brozka 23/19, 43-400 Cieszyn.
3. Poszukuję Geos 2.0 i Deskop 2.0 na C 64, oraz programów: Geowrite 2.1, Geopaint 2.0, Geopublish 2.0, Geofont 2.0. W. Szewczyk, ul. Na Uboczu 14/26, 02-791 Warszawa.
4. Sprzedam C 64 (gwarancja) i joystick. P. Szyderski, Warszawa, tel. 171283.
5. Sprzedam C 64 II, monitor kolorowy 1802D z filtrem (gwarancja), cartridge, 2 joysticki. J. Widziszowski, ul. Olimpijska 20, 44-109 Gliwice, tel. 342702.
6. Sprzedam C 64 II, monitor, Final III, magnetofon, literaturę, akcesoria. J. Kuberacki, ul. Lisia 59/15, Zielona Góra, tel. 65740.
7. Sprzedam C 64, Datasette 1535 (gwarancja), Black Box, joystick. Z. Kułakowski, Zamość, tel. 5452.
8. Sprzedam C 64, Datasette, joystick, literaturę. R. Pawlikowski, ul. Grunwaldzka 26/26, 72-600 Świnoujście.
9. Sprzedam C 64, Final II, magnetofon, 2 joysticki (2.1mln.). P. Szwajca, ul. Radomska 97/21, 25-412 Kielce.
10. Sprzedam C 64, magnetofon, cartridge, joystick (1.8 mln). M. Woźniak, 43-340 Kozy 560.
11. Sprzedam Commodore 128, magnetofon, 3 joysticki, 2 cartridge, literaturę. G. Dąbrowski, 57-251 Laski 32.
12. Sprzedam Commodore C+4. K. Niczyporuk, ul. Wysockiego 18/164, 03-371 Warszawa.
13. Sprzedam moduł pamięci 16KB do Commodore VC 20 (120 tys.). K. Żyła, ul. Reja 2/2, 59-300 Lublin.
14. Sprzedam stację dysków 1541-II do C 64 i 30 dyskietek. M. Sołtykiewicz, 27-400 Ostrowiec Św, tel. 21470.
15. Sprzedam tanio C 64, magnetofon, cartridge, 2 joysticki, literaturę (gwarancja). M. Kowalczyk, ul. Janowskiego 54/37, 02-784 Warszawa, tel. 6416383.
16. Sprzedam tanio Commodore +4 z magnetofonem. A. Zonenberg, ul. Sobieskiego 119, 84-230 Rumia.
17. Sprzedam tanio Commodore VC 20, magnetofon i urządzenia peryferyjne. T. Szulejewski, ul. Goławicka 3/49, 03-550 Warszawa
18. Tanio sprzedam C 64 z monitorem, stacją 5.25", joystickami i literaturą. D. Gajkowski, ul. Aleksandry 23/102, Kraków.

IBM

1. Kupię używanego IBM PC/XT. K. Szmigiero, Warszawa, tel. 402015
2. Kupię wirusy, opisy zarezerwowanych przerwań na IBM PC. S. Fischer. ul. Zamenhofa 26, 60-965 Poznań.
3. Sprzedam IBM XT (NIXDORF) 4.77MHz, 640KB RAM, FDD 360KB, HDD 21MB, CGA, monitor 12" amber, klawiatura, zegar. R. Rauchfleisz, ul. Słowackiego 95/8, 87-100 Toruń.
4. Sprzedam PC AT (12MHz, 2*FDD 1.2MB/360KB, Hercules, monitor 5 mln. R. Trojaczek, ul.Gomułki 8/14, 76-200 Słupsk.
5. Wymienię programy Shareware na IBM. J. Janas, ul. Racławicka 152/2, 02-117 Warszawa, tel. 6590420.

Spectrum

1. Kupię drukarkę do ZX 48 KB. Z. Burgielski, ul. Sanatoryjna 11/a, 58-530 Kowary.
2. Kupię ZX Spectrum 128 KB. R. Podgórski, ul. Legionów 2B/8, 41-200 Sosnowiec.
3. Poszukuję schematu połączenia odbiornika KF z komputerem Timex do programu G1 FTU, RTTY, SSTY. M. Zaława, ul. 3 Maja 35/3, 41-200 Sosnowiec.
4. Sprzedam pojedynczą FDD 3000. J. Wesołowski ul. Gagarina 3/30, 93-530 Łódź, tel. 819997 (wieczorem).
5. Sprzedam przystawkę muzyczną do Timex'a AY-3-8910 (gwarancja). M. Droba, ul. Ćwiklińskiej 2/211, 35-953 Rzeszów, skr. poczt. 248
6. Sprzedam Spectrum+ z klawiaturą SAGA 1. Kupię nowe Spectrum 128 +3. M. Bartnik, ul. Conrada 14/13 01-922 Warszawa.
7. Sprzedam Spectrum+, interface: AY, Kempston, joystick, magnetofon, literaturę. K. Kwiatkowski, ul.Jabłoni 8/23, 38-500 Sanok.
8. Sprzedam Spectrum, FDD 3000, monitor, joystick, literaturę. M. Anders. Kalisz. tel. 31063.
9. Sprzedam stację Polbrit FDD 5000, do ZX Spectrum, dwa japońskie napędy 5.25", kompatybilną z Timex FDD 3000. M. Szefer, ul. Jagiellońska 16b/3, 58-560 Jelenia Góra, tel. 51547.
10. Sprzedam Timex'a 2048 magnetofon i turbo. P. Kozak, ul. Lessla 3/23, 24-100 Puławy, tel. 71307.
11. Sprzedam Timex'a 2048, stację 5.25", AY 3-8910, joystick. A. Wencka, ul. Chłapowskiego 14/4, 63-101 Śrem.
12. Sprzedam Timex'a 2048-80 KB (1.3 mln), FDD 360KB Panasonic-nowy (600 tys.). P. Iwańczuk, ul. Namysłowska 6/12, 03-455 Warszawa.
13. Sprzedam Timex'a 2048. T. Wiśniewski, ul Baczyńskiego 58, Łódź, tel. 560734.
14. Sprzedam Timex'a 2068, magnetofon, joysticki, interface (2 mln.). T. Dawidowski, ul. Okrzei 26, 05-220 Zielonka.
15. Sprzedam Timex'a z magnetofonem. K. Kukla, ul. Łąkowa 4a/1, 47-400 Racibórz.
16. Sprzedam ZX Spectrum 48 KB, interface, 3 joysticki (2 nowe), 3 folie (nowe). M. Łakomy, os. Łokietka 1j/98, 61-616 Poznań, tel. 227518.
17. Sprzedam ZX Spectrum +, magnetofon MK 433, joystick monitor TWM-315. R. Burda, ul. Cegielnlana 24/32, 30-404 Kraków.



Na listy czytelników odpowiadają redaktorzy „Bajtka”

Czy Commodore 64 może współpracować ze stacją dysków FDD 3000? Jeżeli tak to gdzie mogę nabyć interface?

Jakub Cebula

Niestety niemożliwe jest podłączenie tej stacji do Commodore 64 między innymi z dwóch powodów:
1. Niezgodność systemowa. Stacja dysków C64 pracuje w systemie DOS 2.6, a SPECTRUM (FDD 3000) w systemie TOS.
2. Niezgodność sprzętowa. Znaczne różnice pomiędzy sterownikami w stacjach C64 i SPECTRUM.

(PLis)

✳

Pilnie poszukuję użytkownika interfejsu DISCIPLE firmy ROCKFORT do ZX Spectrum. Mój odmówił posłuszeństwa i nie mam dostępu do materiałów na dyskietkach 3.5”.
Będę wdzięczny za kontakt „pod” tel. 270319 wieczorem lub 219460 do godziny 14.

Piotr Michałowski

Drukujemy ten list, gdyż nie byliśmy w stanie pomóc panu Piotrowi. Mamy nadzieję, że znajdzie się ktoś, kto będzie w stanie pomóc.

✳

Od kilku miesięcy posiadam komputer Amiga 500. Mam kilka pytań odnośnie możliwości graficznych Amigi:
1. Jakie są poszczególne tryby rozdzielczości ekranu?
2. Po ile kolorów można w każdym z tych trybów uzyskać?
3. Co to jest tryb Extra Half Bright?
4. Dlaczego obraz w najwyższej rozdzielczości miga?
5. Ile Amiga może wyświetlać sprite'ów?

1. Można wyróżnić cztery tryby rozdzielczości:
— 320*256 punktów,
— 640*256 punktów,
— 320*512 punktów,
— 640*512 punktów

Należy przy tym zaznaczyć, że podane rozdzielczości nie uwzględniają całego ekranu, pomijają bowiem ramkę. Amiga nie wyróżnia co prawda ramki, tak jak to robi na przykład Atari ST, ale nie ma sensu wykorzystywać całego ekranu, gdyż na większości monitorów brzegi ekranu nie byłyby widoczne, są też z reguły mniej wyraźne. Wyświetlanie całego ekranu nie jest co prawda żadnym specjalnym trybem, nadano mu jednak nazwę OVERSCAN. Maksymalna rozdzielczość Amigi to wtedy około 700*600 punktów

2. Amiga może wyświetlać 2, 4, 8, 16, 32, 64 lub 4096 kolorów na ekranie na raz. Tryby 2 do 16 kolorów dostępne są w każdej rozdzielczości, natomiast 32, 64 i 4096 tylko przy niskiej rozdzielczości poziomej, czyli 320*256 i 320*512 punktów. Spowodowane jest to koniecznością przesłania w trybie wysokiej rozdzielczości poziomej dwa razy większej ilości danych w tym samym czasie, co przerastałoby możliwości układów Amigi przy większej liczbie kolorów.

3. EHB to tryb, w którym Amiga może wyświetlać do 64 kolorów naraz na ekranie. W trybie tym Amiga tworzy dodatkowe 32 kolory (ma tylko 32 rejestry kolorów) przez ściemnienie o połowę pierwszych 32 kolorów. Na punkt ekranu przypada wtedy 6 bitów — pięć określa jeden z 32 kolorów, szósty — czy dokonać ściemnienia.

4. Dla ścisłości — to w trybie INTERLACE, czyli przy podwojonej liczbie linii, obraz miga. Taka jest już niestety konstrukcja Amigi. Mechanizm jest dokładnie taki sam jak w telewizji (zauważyłeś z pewnością migotanie np. na podpisach w „Wiadomościach”) — poszczególna linia jest wyświetlana tylko 25 razy na sekundę, a przy tej częstotliwości oko ludzkie dostrzega już migotanie. Linie parzyste są wyświetlane na przemian z nieparzystymi. W ten sposób obraz jest tworzony co 1/50 sekundy, ale dla odebrania pełnej rozdzielczości potrzeba 1/25 sekundy. Przyjęcie takiego rozwiązania umożliwiło zredukowanie ilości przesyłanych danych o połowę, odbiorniki były tez dzięki temu tańsze.

5. Amiga może wyświetlać do ośmiu sprite'ów, każdy w czterech kolorach. Sprite'y można też łączyć po dwa, dzięki czemu mogą one mieć osiem kolorów. Sprite'y mogą mieć do 16 punktów szerokości przy dowolnej wysokości. Istnieje też możliwość zdefiniowania sprite'a od nowa zaraz po jego wyświetleniu, a przed końcem wyświetlania ekranu (ekran jest rysowany 50 razy na sekundę od góry do dołu). W ten sposób można liczbę sprite'ów na ekranie wielokrotnie pomnożyć.

Andy

✳

Dzięki świeżemu napływowi gotówki zdecydowałem się na zakup komputera Commodore 64. Nie mogę się jednak zdecydować w jakiej konfiguracji zakupić ten sprzęt. Czy z magnetofonem, czy lepiej ze stacją dysków?

Paweł Niemojski

Kupując sprzęt firmy Commodore należy się liczyć z pewnymi wydatkami na urządzenia peryferyjne. C64 jest przystosowany do współpracy ze specjalnym magentofonem. Magnetofon jest stosunkowo najtańszą lecz jednocześnie wolniejszą i uciążliwą pamięcią masową komputera. Na taką opinię składa się kilka faktów:
— wolne oraz niewygodne wyszukiwanie programów zapisanych na kasecie magnetofonowej,
— problemy związane z różnicą ustawienia skosów głowicy, a w efekcie trudności w odczytaniu programów nagranych z pomocą innego magnetofonu,
— powstaje coraz więcej programów, które wymagają stacji dysków.

Rzecz się ma zupełnie inaczej w stosunku do stacji dysków. Zapis danych na dyskietce jest pewniejszy niż w przypadku magnetofonu. Ponadto swoją rolę odgrywa także szybkość pracy stacji dysków w stosunku do magnetofonu. Jednak aby uzyskać znaczne przyspieszenie niezbędne jest zastosowanie modułu (ang: cartridge) np. FINAL II, III lub ACTION REPLAY. Moduły takie oprócz przyspieszenia urządzeń zewnętrznych zawierają także wiele innych pożytecznych funkcji jak np. monitor języka maszynowego, rozszerzenie języka BASIC czy opcję druku ekranu na drukarce.

Zdecydowanie polecam więc stację dysków także ze względu na fakt iz programy zajmujące całą dyskietkę lub więcej, ewentualnie komunikujące się z dyskiem niestety nie będą pracowały z magnetofonem. Należy zwrócić także uwagę, że magnetofon jest już przeżytkiem, a więc swego rodzaju pułapką na oszczędnych.

Sądzę, że rozwiałem wszelkie wątpliwosci odnośnie wyboru urządzenia współpracującego z Commodore 64, a odpowiedzialnego za przechowywanie danych. Życzę trafnego wyboru

PLI

✳

Szanowna Redakcjo!
Nawiązując do artykułu „Teletext na Commodore” (Bajtek 3/91) chciałbym nadmienić, że autor powyższego artykułu myli się, jakoby na rynku zachodnim był tylko dostępny jeden typ urządzenia do odbioru teletekstu. Ja znam osobiście następne dwa.
Jeden z nich sprzedawany jest przez firmę WIEGAND VIDEODATEN SYSTEME w dwóch wersjach:
VT 201 — standardowy dekoder wyposażony w RS-232 (349 DM),
VT 232 — dekoder profesjonalny z dużą liczbą funkcji dodatkowych (549 DM).
Do obu urządzeń dodawane są dodatkowo programy dla MS-DOS, Atari lub Amigi. Oba dekodery pozwalają na odbiór teletekstu ze wszystkich nadających się programów. Następnie można tak odebrane dane zapisać i wydrukowac. Dzięki dodawanym programom można odtworzyć 2000 stron teksty z możliwością ich prezentacji w formie graficznej lub tekstowej (ASCII).
Drugi dekoder [...] wytwarzany we Włoszech przez ECR Elettronica [...] nosi nazwę Televideo MK2 Teletext-Videotext-Decoder (cena nie jest mi znana). Przeznaczony jest on dla Amigi, załączane są także programy na dyskietce. Jego zalety to łatwa obsługa, pełny dekoder, możliwość zapisywania stron teletekstu. Wadą natomiast jest brak przewodu połączeniowego z magnetowidem.

Andrzej Siciński „KANGU”
Wiesbaden

Od redakcji: Dziękujemy za list!

Od pół roku mam Schneidera 6 * 464. Czytając „Bajtka” nie zauważyłem żadnego artykułu na temat mojego komputera. Chciałbym więc zadać kilka pytań
1. Czy programy pisane na CPC 6128 mogą działać na CPC 464?
2. Czy można na CPC 464 używać CP/M Plus?
3. Jak podłączyć do Schneidera dwa joysticki? Pisaliście, że można i to samo jest w instrukcji, a jest tylko jedno gniazdko.

J. Małkowski

1. Programy pisane na CPC 664 i CPC 6128 mogą działać na CPC 464, jeśli nie wymagają stacji dysków (to zastrzeżenie nie dotyczy, oczywiście 464 rozszerzonego o interfejs DD1 i stację) i nie wymagają dodatkowej pamięci (to dotyczy programów na 6128). Kłopoty mogą być czasem związane z błędami w BASIC-u 464.
2. Niestety nie. CP/M Plus wymaga stacji dysków i 128K pamięci. Po podłączeniu stacji można używać CP/M 2.2 — wielu programom to wystarczy.
3. Można, wymaga to przejściówki. Schemat przejściówki będzie opublikowany w numerze 11 lub 12. Jeśli nie możesz czekać — poszukaj w starych numerach „Bajtka”, drukowaliśmy już kiedyś schemat takiej przejściówki — niestety z błędem uniemożliwiającym jednoczesne używanie obu joysticków.

(MSZ)

ZWYCIĘZCY!

Oto szczęśliwi (i dokładnie czytający „Bajtka") uczestnicy Wielkiego Konkursu „7 Pytań" z numeru majowego:

— **niespodziankę** — komputer Commodore C-64 — zdobył Wojciech Michalski z Radości:
- joystick MEGABOARD wylosował Jarosław Szaciłło z Warszawy
- joystick SUPERBOARD przypadł Dariuszowi Śliwie z Kielc
- joystick TOP STAR zdobył Edward Galikowski z Janikowa
- joysticki JUNIOR wylosowali: Sławomir Migas, Marcin Kierklo, Daniel Salostowicz, Bartłomiej Karczewski, Mariusz Jaworski
- joysticki JUNIOR STICK zdobyli: Sławomir Majcher, Paweł Moll, Radosław Naremski, Maria Wiśniewska, Kamil Dobrzyński, Radosław Olkowicz, Janusz Guttmeyer, Dariusz Przeszłowski, Dariusz Łyczek i Piotr Jacuński
- pudełka na dyskietki 5.25" wygrali: Mieczysław Szefer, Artur Szmidt, M. Janicki, Piotr Nowakowski, Marcin Mikulski, Konrad Więctowski oraz Jacek Kabawiński
- pudełka na dyskietki 3.5" przypadły: Michałowi Rosochaczowi, Tomaszowi Stolarczykowi i Cezariuszowi Guthowi
- po 10 dyskietek 5.25" otrzymują: M. Laskowski, Piotr Pezacki, Marcin Kot, Piotr Jasiniewski i Andrzej Wydrychiewicz
- 10 dyskietek 3.5" zdobył Piotr Misztal
- dyskietki czyszczące otrzymują: 5.25" — Błażej Ciesielski, 3.5" — Paweł Kopyt.
- podkładkę pod mysz zdobył Olaf Pietrewicz.

A teraz poprawne odpowiedzi — kto napisał poprawnie, może się dowartościować:

Najwięcej firm na targach CeBIT było z Niemiec. Złoty podział to metoda podziału odcinka. Najnowsza wersja Turbo Pascala to „Turbo Pascal for Windows". Big Blue to IBM (większość odpowiedzi — Commodore, źle!). Artykuł pt. „SHELL" powstał w grudniu 1990 (data była na wydruku). EOF to znak końca pliku (End Of File). Znak o kodzie 155 oznacza dla Atari XL/XE nowy wiersz.

(MSZ)

7 PYTAŃ
SIERPIEŃ '91

Głównymi nagrodami są JOYSTICKI ufundowane przez firmę TAL sp.z.o.o. o łącznej wartości "jedynie" 4 milionów! Dodatkowo - NIESPODZIANKA ufundowana przez Sklepy "Bajtka".

Wystarczy tylko odpowiedzieć (ale poprawnie!) na pytania konkursowe, stawiając krzyżyk przy jednej z podanych odpowiedzi i wysłać kupon (oryginał, nie ksero!) na adres "Bajtka" do dnia 30 września - decyduje data nadejścia! Nie zapomnij o dopisku "KONKURS '7 PYTAŃ' - SIERPIEŃ'91".

NAGRODY:
- **NIESPODZIANKA**

☐ **GRUPA I**
- **10 Junior Sticków**
- **5 Juniorów**
- **1 Top Star**
- **1 Megaboard**
- **1 Superboard**

☐ **GRUPA II**
- **5 pudełek dyskietek 5.25"**
- **7 pudełek na 80 dyskietek 5.25"**
- **dyskietka czyszcząca 5.25"**

☐ **GRUPA III**
- **1 pudełko dyskietek 3.5"**
- **3 pudełka na 80 dyskietek 3.5"**
- **dyskietka czyszcząca 3.5"**
- **podkładka pod mysz (mouse mat)**

Spółdzielnia
"Bajtek"
ul. Wspólna 61
00-687 Warszawa

MIEJSCE NA ZNACZEK

KUPON PRENUMERATY
AKTUALNY DO **30.09.1991**

Co miesiąc kolejny zaktualizowany kupon.

liczba kol. zeszytów	3	6	12	po ile egzempl.
Bajtek	X	60000	120000	
Moje Atari	24000	48000	X	
TOP SECRET	27000	54000	X	

WPŁAT DOKONYWAĆ NA KONTO

Spółdzielnia "BAJTEK"
Bank "Agrobank S.A."
470005-1834-131
ul. Grochowska 262
04-398 Warszawa

Wytnij lub zrób kserokopię i przyślij do nas.

Znowu cos do wygrania! Samych joysticków za ponad 4 miliony, nie mówiąc o NIESPODZIANCE (pewnie znowu C-64, ale kto wie?). Żeby coś wygrać trzeba jednak odpowiedzieć na siedem prostych pytań. Potem trzeba wyciąć kupon, nakleić na kartkę pocztową i wysłać - im szybciej tym lepiej.
A teraz do dzieła!

1. Co znaczy skrót BSA ?
- ☐ Bad Sector Association
- ☒ Business Software Alliance
- ☐ Big System Applications
- ☐ Back Shot Action

2. Ile znaków mieści Clipper w polu znakowym ?
- ☐ 256
- ☐ 1024
- ☐ 16000
- ☒ 32768

3. Ile kosztuje elektrownia atomowa w grze SimCity ?
- ☐ 3000 $
- ☐ 4000 $
- ☒ 5000 $
- ☐ 6000 $

4. Ile pamięci ma SAM Coupe ?
- ☐ 64 K
- ☐ 128 K
- ☒ 256 K
- ☐ 512 K

5. Ile baz danych moze jednocześnie obsługiwać dBase II ?
- ☐ jedną
- ☒ dwie
- ☐ cztery
- ☐ dziesięć

6. Co to jest RUSHMORE ?
- ☐ gra firmy Sierra On-Line
- ☐ obiektowy język programowania
- ☒ algorytm szybkiego wyszukiwania danych
- ☐ system operacyjny komputera Cray XMP-3

7. Co będzie tematem 12. odcinka "Języka maszynowego" ?
- ☐ rozkazy prefiksowane przez #ED
- ☒ opisy procedur z ROM-u
- ☐ metody korzystania z przerwań
- ☐ morderstwo Alexis Colby

IMIĘ I NAZWISKO: ______________________
ADRES: ______________________
______________________ GRUPA: ______

UWAGA! Odpowiedzi otrzymane po 31 września, odbite na ksero (lub w inny sposób) lub nie posiadające dopisku - NIE WEZMĄ UDZIAŁU W LOSOWANIU NAGRÓD!

imię nazwisko.........................
ulica, nr.........................
kod, miejscowość.........................
numer prenumeratora.........................

- Prenumerata zawarta przed upływem ważności kuponu gwarantuje niezmienność cen
- Przesyłka pocztowa nie wymaga dodatkowych opłat
- Minimalny czas realizacji zamówienia 4-6 tyg.
- Jeżeli w ciągu 2 tyg. od pojawienia się numeru w kioskach nie nadeszła przesyłka, redakcja prosi o kontakt
- Za błędy wynikające z niestarannego wypełnienia formularza redakcja nie ponosi odpowiedzialności
- Prosimy o wyraźne zakreślenie odpowiednich ilości egzemplarzy w tabeli

TU WKLEIĆ ODCINEK PRZEKAZU
(potwierdzenie dla wpłacającego)

szukaj znaku TAL tam znajdziesz wszystko!

DDD®

DOSTAWA DO DOMU KURIEREM

Max 24 godziny od złożenia zamówienia bez względu na miejsce zamieszkania. POWIEDZ TO KOLEDZE, KTÓRY NIE MA KOMPUTERA.
PŁATNOŚĆ PRZY ODBIORZE.

NASZA OFERTA:

ATARI:

ATARI 65 XE + magn.
ATARI 130 XE + magn.
Stacja dysków do ATARI
ATARI 1040 STFM + monitor
ATARI 1040 STE + monitor
Stacja dysków do ATARI ST
Mouse Pad

AMIGA:

AMIGA 500 (klaw. ang. wers. 132)
AMIGA 2000
Stacja dysków do AMIGI
Rozszerzenie pamięci 512 KB
Modulator do AMIGI
Monitory COMMODORE, PHILIPS

COMMODORE:

COMMODORE C-64 II + magn.
COMMODORE C-64 VGS + magn.
Stacja dysków do C-64
Monitory COMMODORE, PHILIPS

DODATKI:

Dyskietki 3,5" — N.N. JVC, KODAK, Nashua, Edixa
Dyskietki 5,25" — N.N. JUC, KODAK, Nashua, Edixa
Instrukcje w języku polskim
Wycinarki
Scanery
Filtry ochronne

Gwarancja 12 miesięcy

Aktualne informacje o cenach sprzętu można uzyskać dzwoniąc pod numer 23-92-21 lub kupując „Gazetę Wyborczą", wydanie sobotnio-niedzielne, strona z programem telewizyjnym. W przypadku sprzętu o wartości powyżej 1.000.000 zł należy przesłać przedpłatę 300.000 zł na nasz adres (przekazem **telegraficznym** lub zwykłym). Reszta należności płatna kurierowi przy odbiorze.

NASZE CENY SĄ NISKIE

**Adres firmy: „TAL", ul. Mikowa 45,
02-411 Warszawa Włochy
dojazd autobusem 173 z Pl. Narutowicza
lub PKP do stacji W-wa Włochy**

**tel.: 23-86-83 sp. hurtowa
23-92-21 interwencje
Fax: 23 88 34
godz. pracy:
pon.–pt. 9–17
sobota 9–16**



SV 119
Junior
2 Fire
6 Blaszanych styków
Prosty mechanizm
69 900 zł



SV 122
Quickjoy II
2 Fire
6 Blaszanych styków
AutoFire
Drążek lotniczy
89 900 zł



SV 124
Turbo
6 Mikrostyków
AutoFire
Drążek lotniczy
109 900 zł



SV 126
Jet Fighter
2 Fire
6 Mikrostyków
AutoFire
ACS-Regulator
szybkości AUTO
Obsługa pod kciuk"
Drążek lotniczy
169 900 zł

SV 201
Wersja do IBM
249 900 zł
+ SV 210
Game Card
249 900 zł



SV 128
Megaboard
4 Fire
10 Mikrostyków
AutoFire
6 cyfrowy stoper
ATM — Anti Tilt Mechanism
Fire Pad
Sygnał dźwiękowy
zwalnianie pracy
komputera
Cyfrowy
wyświetlacz
czasu
319 000 zł



SV 125
Superboard
6 Fire
10 Mikrostyków
AutoFire
Cyfrowy wyświetlacz
czasu
Sygnał dźwiękowy
Przełacznik dla
leworęcznych
Drążek lotniczy
259 900 zł

NR 1 NA ŚWIECIE!
Wytrzymuje min. 2 lata !!!



SV 127
Top Star
2 Fire
6 Mikrostyków
AutoFire
Przezroczysta obudowa
SAS — Shock Absorbing
System
Platynowane części
Zwalnianie pracy
komputera
269 000 zł

SV 227
Wersja do IBM
269 000 zł
+SV 210
Game Card
249 900 zł



SV 123
Supercharger
2 Fire
AutoFire
6 Mikrostyków
Ergonomiczna budowa
Precyzyjny mechanizm
119 900 zł



SV 202
M 6 analog
Analogowy
DO IBM XT/AT
(kompatybilnych)
Współpracuje z Game-Card
lub I/O Card
2 Fire
229 900 zł



Van 5
Pudełko na dyskietki
80 sztuk 5¼"
Zamknięcie na klucz
119 900 zł

JAK KUPIĆ JOYSTICKI?

Daną kwotę należy przesłać na adres firmy zwykłym przekazem pocztowym (czerwony blankiet), podając na odwrocie (w miejscu na korespondencję) **dokładny adres oraz symbol** joysticka. Czas realizacji zamówienia 8 dni.

10% zniżki na zamówienia na joysticki złożone do 30 września!

TEST JOYSTICKÓW: Bajtek 1/91, TOP SECRET 3/91

Quickjoy

TAL

BZYKOMANIA

Już w dawnych czasach zajmowano się takimi nietypowymi zjawiskami jak anomalie; jak wiadomo z życia, prowadzą one do rozmaitych wypaczeń, ale jednocześnie i odkryć. Olbrzymia większość anomalii w naszym życiu codziennym jest niestety autorstwa naszych bliźnich, a przynajmniej ich określonej grupy zwanej potocznie „rzutkimi umysłami"



I tak na przykład niejaki Bańbuła odkrył, że zalecana przez rozmaite kodeksy jazda automobilem pomiędzy dwiema namalowanymi na jezdni liniami jest zdecydowanie szkodliwa. Do tego niewątpliwie sensacyjnego odkrycia przyczyniły się montowane w najbardziej idiotycznych miejscach jezdni studzienki ściekowe i związana z tym konieczność regularnej wymiany całego zawieszenia dwa razy na kwartał. Teraz Bańbuła jeździ środkiem szosy, bo wyliczył bez komputera, że mandaty mniej go kosztują niż zawieszenie.

Rzutkie umysły działają także z zaciekłością w dziedzinach typowo komputerowych. Oto ostatnio coraz więcej słychać o modemach. Jest to niewątpliwie następna anomalia, jako że powinno być słychać więcej PRZEZ modemy; dopóki jednak rodzima telefonia nadal będzie korzystała z genialnych pomysłów szeregu, jak przypuszczam, rzutkich umysłów, do tej pory będziemy więcej słyszeć o modemach, a nie przez nie.

Zwiększająca się ilość podań o rejestrację modemu spowodowała powołanie grupy ekspertów z docentem Nielotnym na czele do zbadania tego podejrzanego zjawiska. Zachodziło przypuszczenie, że grupa technokratów chce wybzykać się modemem z opłat abonenckich. Po wszechstronnym zbadaniu sprawy docent napisał pracę profesorską, w której udowodnił, czarno na białym, że:

a) modem zużywa rodzime linie telefoniczne znacznie bardziej aniżeli wrzaski abonentów,
b) modem stanowi poważne zagrożenie dla krajowej telefonii, jako że bzykanie w linię powoduje wibrację drutów na słupach,
c) modem może przesłać więcej informacji w jednostce czasu i fakt ten coś niewątpliwie oznacza,
d) punkt c) powyżej pozwala na wprowadzenie do taryfy usług nowej pozycji, za którą pobierana będzie opłata w wysokości absolutnie dowolnej, bo niby dlaczego ktoś, kto zajmuje krócej linię telefoniczną, miałby mniej płacić (liczniki impulsowe!),
e) normalna rozmowa telefoniczna i transmisja danych to zupełnie dwie różne rzeczy. Fakt, że przesyłanie informacji odbywa się po tych samych łączach i nawet w podobnym zakresie częstotliwości nie ma żadnego znaczenia; bzykanie w słuchawkę to jest jednak bzykanie i należy w związku z tym przystąpić do opracowywania cennika dla bzykaczy uwzględniającego inflację w roku 17453.
f) należy zorganizować brygady specjalne (mo)demolicji zajmujące się wyłapywaniem takich cwaniaczków, co to tylko chcieliby sobie bzyknąć w linię i odłożyć słuchawkę.
g) wprowadzenie modemów do powszechnego użytku spowoduje masowe bezrobocie zwłaszcza wśród grupy sanitarno-epidemiologicznej, zajmującej się wpsikiwaniem podejrzanych substancji do słuchawki.

Raport docenta Nielotnego wstrząsnął do głębi niektórymi dostojnikami uświadamiając im grozę sytuacji i powagę niebezpieczeństwa. Z drugiej jednak strony pocieszyli się oni faktem (niezaprzeczalnym), że jakość łączy i linii jest tak zaawansowana, iż zupełnie niemożliwe jest jej dalsze obniżenie. Jako zagrożenie główne przyjęto stosowanie modemów razem z techniką impulsów telefonicznych — 15 minut gadania (za co lecą sobie pieniążki) można zastąpić minutową transmisją modemową i w ten sposób kasa z dnia na dzień pustoszeje. Żaden jednak z mędrców nie wpadł na pomysł, aby zarazić krajowy przemysł elektroniczny (wyśmienicie prosperujący po ostatnich reformach finansowo-podatkowych) ideą taniego modemu dla użytkowników komputerów, który można by na przykład kupować od poczty wraz z abonamentem.

W niektórych krajach pospólstwo wymyśliło, że modem może być całkiem użyteczny i w związku z tym parę firm założyło szereg dużych banków danych, do których można się nawet dodzwonić. Do przepastnych pamięci tychże banków wprowadzono informacje o numerach telefonicznych, rozkładach jazdy różnych pojazdów wodno-powietrzno-lądowych, reklamy, gazety, biura porad, telefony zaufania i szereg rozmaitych innych danych, jakie mogą się przydać dosłownie każdemu. Wyobraźmy sobie, że jesteś zainteresowany udaniem się na wycieczkę na wyspy Bahama; dzięki modemowi i takim właśnie bankom danych możesz uzyskać kupę szczegółów o planowanej trasie, datach, rezerwacjach i cenach nie ruszając się z domu. Dostęp do banku oczywiście kosztuje (miesięcznie kilkanaście dolarów plus rachunek za telefon), ale po takim seansie okazuje się zwykle, że zapłaciłbyś więcej za paliwo zużyte na dojazdy do poszczególnych biur podróży. Jednym słowem modem *tam* oznacza dostęp do świata informacji, w Polsce zaś — ćwiczenie cierpliwości w całym znaczeniu słowa „cierpliwość".

Czy warto kupić modem? Chyba tak. W kraju istnieje już szereg lokalnych BBS i choć nie ma w nich rozkładu jazdy PKP, to już za jakieś 500 lat może ktoś o tym pomyśli. Komunikacja np. z USA może się udać pod warunkiem, że tamtejszy komputer zrozumie początek polskiego protokołu transmisji (hasło „Mówi się czy nie?!"). Choć połączenie z Włocławkiem wykończyło mnie doszczętnie, jest już zauważalny postęp — kiedyś słyszałem 147 abonentów naraz, a wczoraj tylko 144. Oczywiście rozmówcy nie słyszałem w ogóle.

Druga strona medalu to niezwykle wprost ułatwienia oferowane amatorom modemów ze strony rozmaitych placówek telepocztowych. Modem musi być homologowany, co jeszcze jestem w stanie zrozumieć. Posiadacz rzeczonego urządzenia musi zarejestrować się w odpowiednim urzędzie, bo jak (mo)demolicja namierzy klienta, że zamiast gadać — bzyczy, to natychmiast zabiorą mu i telefon, i numer (ciekawe jak podsłuch linii telefonicznych jest obecnie usankcjonowany prawnie i jakim cudem można odróżnić bzyczenie modemu od bzyczenia samej linii). Oczywiście jeśli burżuja stać na modem, to dlaczego nie miałby on płacić za to zupełnie oddzielnie — z czegoś przecież trzeba utrzymać Departamenty Zwalczania Abonentów czy rejestrację „bzykaczy". W zamian za te wszystkie udogodnienia możesz usiąść sobie wieczorem i pozastanawiać się, gdzie by się tu dobzykać w następnej pięciolatce.

Ja bym wprowadził jeszcze parę usprawnień. Na przykład: posiadaczowi modemu wolno bzykać tylko w piątki od 16.00 do 16.10. Albo: bzykanie w Warszawie jest zabronione. Jako rzetelny wymiar stawki rejestracyjnej za modem można przyjąć formułę:

liczba bodów * numer telefonu = opłata rejestracyjna

Ponieważ docent Nielotny nie wpadł na ten pomysł, profesura przeszła mu koło nosa; dostał ją magister Zdolniacha za projekt ustawy o obowiązkowym pisemnym przedstawianiu i zatwierdzaniu w rejonowej CIA (Centrali Inwigilacji Abonentów) danych przeznaczonych do wysłania modemem.

Zdrowy rozsądek tłamszony nadal przez rozmaite „rzutkie umysły" podpowiadałby inne rozwiązania, choćby założenie przez instytucje zarządzające liniami telefonicznymi banku danych dla amatorów modemu, czy opracowanie i sprzedaż własnych konstrukcji do najpopularniejszych w kraju komputerów. W niektórych krajach wykazano (to są pewnikiem jakieś kanty), że to się opłaca. U nas — nie. Podobnie jak ochrona programów.

Ja nie mam problemu. Jestem na liście wiecznie oczekujących i telefonu spodziewam się koło roku 14973. Czyli mam spokój i pobzykać mogę sobie wyłącznie po wsadzeniu pszczoły do słuchawki.

Klaudiusz Dybowski

P.S. Nazwiska osób wymienionych w artykule są fikcyjne.